CONNECTED SERVICE FRAMEWORK

CONNECTED SERVICES FRAMEWORK

DOCUMENT	Master Document
VERSION	v2.0
DATE	June 2026
CLASS	Public
AUDIENCE	MAPs · CPs · HMAPs · regulators · integrators · industry bodies
STATUS	Published

EXECUTIVE SUMMARY

This is the canonical single-volume publication of the Connected Services Framework (CSF) — an open, peer-to-peer transport standard developed by the Telecom Technical Architecture Group (TAG) for the UK telecommunications industry. The Master Document combines the framework introduction with Part 1 (Framework — for decision-makers and architects) and Part 2 (Implementation — for developers and integration engineers) into one navigable volume with internal cross-references that work end-to-end. Standalone companion documents (Definitions & Terminology, the CP Rights Charter, Onboarding & Testing Process, TOTSCo HMAP Integration Guide, the FAQs, and the TAG governance documents) are published separately and are referenced from this volume.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Part 1 — Framework (Management & Architecture)

For decision-makers, solution architects, and programme managers — the strategic and architectural view of the CSF.

1. Introduction 20
1.1 Purpose of This Document 20
1.2 What the CSF Does 20
1.3 Who This Document Is For 20
1.4 Scope 21
1.5 The Case for a Distributed Model 21
1.6 Initial Application: Switching for Business (SforB) 22
1.7 Document Structure 23
2. Background 24
2.1 The Evolution of UK Telecoms Switching 24
2.2 One Touch Switch (OTS): Establishing the Foundation 24
Limitations of the OTS Implementation 24
2.3 Switching for Business (SforB): Removing Restrictions 25
2.4 The Telecom Technical Architecture Group (TAG) 25
2.5 From OTS to CSF: What Changed 26
3. Principles and Requirements 27
3.1 How Requirements Were Gathered 27
3.2 Requirements 27
Connectivity and Routing 27
Directory Management 28
Messaging 28
Operations 28
Other 28
CP Right to Move 29
3.3 Summary of Principles 29
4. Architecture Overview 30
4.1 Layered Architecture 30
4.2 Entity Definitions 30
4.2.1 Communications Provider (CP) 30

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

4.2.2 Managed Access Provider (MAP) 30
4.2.3 MAP of 1 31
4.2.4 Hub Managed Access Provider (HMAP) 31
4.3 Message Flow 32
Step-by-Step Workflow 32
4.4 CSF Networks 33
4.5 API Versioning and Feature Releases 33
API Versioning 33
Feature Discovery 33
4.6 What the CSF Does Not Prescribe 34
5. CP Registry, Master Registry, and Directory 35
5.1 Overview 35
How the Three Tiers Work Together 35
5.2 CP Registry Structure 36
MAP Section 36
CP List Section 36
5.3 CP Registry Synchronisation 37
Frequency 37
Building the Master Registry 37
Deriving the Directory 37
5.4 Advantages of the Distributed Model 38
5.5 Conflict Resolution 38
5.6 TOTSCo Compatibility 38
6. Security Overview 39
6.1 Security Architecture 39
6.2 Transport Layer Security (TLS 1.3) 40
What TLS 1.3 Provides 40
Why TLS 1.3 Specifically 40
6.3 OAuth 2.0 40
How OAuth 2.0 is Used in the CSF 40
CSF OAuth 2.0 Requirements 41
6.4 API Key Authentication 41
6.5 PKI and DKIM Message Signing 41
Why DKIM? 41
How It Works (Conceptual) 42
DNS Records for Each CP 42
© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

6.6 Security Summary 43
7. Onboarding 44
7.1 Overview 44
7.2 Onboarding a New MAP 44
7.2.1 Sponsorship Framework 44
7.2.2 On-Ramp Process 45
7.3 Onboarding a CP 46
7.3.1 MAP Responsibilities 46
7.3.2 When a CP Can Exchange Messages 46
7.4 MAP-of-1 Onboarding (incl. TOTSCo → CSF Migration) 47
7.4.1 Definition and scope 47
7.4.2 Strategic context — the CP remains in control 47
7.4.3 Decision tree — "Should I become a MAP of 1?" 48
7.4.4 Pre-migration capability checklist 48
7.4.5 Sponsor MAP programme — fast track for single-CP 49
7.4.6 Finding your Sponsor MAP 49
7.4.7 The cutover journey — big-bang, CP-controlled 50
7.4.8 RCPID allocation rule 51
7.4.9 Phase 4 relaxation for MAP-of-1 51
7.4.10 Bilateral testing of additional protocols 52
7.4.11 Cost model — qualitative 52
7.4.12 What a MAP-of-1 *does not* lose 53
7.5 Testing in Production 53
7.6 Memorandum of Understanding (MoU) 53
8. RCPID Management 54
8.1 What is an RCPID? 54
8.2 Why UUIDv4? 54
Problems with the OTS Format 54
Benefits of UUIDv4 54
Compatibility with OTS 54
8.3 Allocation Process 55
8.4 RCPID Lifecycle 55
Allocation 55
MAP Transitions 55
Retirement 56
8.5 Handling Exports During Transitions 56

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

8.6 Summary 56
9. Operational Excellence 57
9.1 Core Principles 57
9.2 Availability Target 57
9.3 High Availability and Redundancy 57
9.4 Continuous Operations During Maintenance 57
9.5 Monitoring and Alerting 57
9.6 Disaster Recovery 58
9.7 Security and Compliance 58
9.8 Guaranteed Message Delivery 58
Pre-Send Verification 58
Retry Policy 58
Asynchronous Error Handling 59
9.9 Service Level Agreements 59
9.10 Capacity Planning 59
What counts as a Switchable Asset 60
10. Governance 61
10.1 Standards Landscape 61
10.2 Standards Bodies 61
10.2.1 OTA2 — Office of the Telecommunications Adjudicator 61
10.2.2 TOTSCo — Telecoms One Touch Switching Company 61
10.2.3 GPLB-SG — GPLB Steering Group 62
10.2.4 TAG — Telecom Technical Architecture Group 62
10.3 Change Control 63
CSF Documentation 63
Industry Process Documentation 63
10.4 Memorandum of Understanding 63
10.5 Decentralised Governance 63
10.6 Governance Controls 64
10.6.1 TAG Decision-Making and Voting 64
10.6.2 MAP Compliance and Periodic Review 64
10.6.3 Version Adoption Policy 64
10.6.4 Dispute Resolution 65
10.6.5 Incident Response Coordination 65
10.6.6 MAP Suspension and Removal 66
10.6.7 Anti-Competitive Safeguards 66

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

10.6.8 CP Representation 67
10.6.9 Documentation and Transparency 67
10.7 Supporting Documents 68
Glossary 69
Abbreviations 69
Key Definitions 70
CP Registry 70
Master Registry 70
Directory 71
Sponsor MAP 71
New MAP 71
Bit pipe 71
SOR (Switch Order Reference) 71
Letterbox 71
Routing Group 71
Feature Release 71

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Part 2 — Implementation (Technical)

For developers, integration engineers, and technical architects — the wire-level reference and the OpenAPI appendix.

1. Getting Started 74
1.1 Purpose 74
1.2 Prerequisites 74
1.3 Implementation Checklist 74
Infrastructure Setup 74
API Implementation 74
Security Implementation 75
Directory and Registry 75
Message Processing 75
Operations 75
Onboarding 75
TOTSCo Integration (if applicable to the protocol e.g. SforB) 76
1.4 Recommended Endpoint Structure 76
1.5 Development Resources 77
2. OAuth 2.0 Implementation 78
2.1 Overview 78
2.2 Access Token Request 78
Request Headers 78
Request Body Parameters 78
2.3 Access Token Response 79
2.4 Using the Access Token 79
2.5 Token Lifecycle 79
2.6 Error Responses 79
2.7 Security Requirements 80
3. PKI and DKIM Message Signing 81
3.1 Overview 81
3.2 Public/Private Key Pairs 81
3.3 DNS Records 81
Domain Key Record 81
MAP Key Record 82
Why Two Records? 82

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

CP Responsibilities 82
MAP Responsibilities 82
3.4 DKIM Signature Process 83
3.5 CSF HTTP Headers 84
X-CSF-SIGNATURE Structure 84
Mandatory Tags 84
Mandatory Tags with Fixed Values 84
Optional Tags with Fixed Values 84
DKIM-Signature Canonicalisation 84
3.6 Supported Algorithms 85
Encryption and Signing 85
Key Length Requirements 85
DNS Key Examples 85
Canonicalisation 85
3.7 Public Key Storage 86
3.8 Worked Examples 86
3.8.1 Java — Creating a Signature 86
3.8.2 Java — Verifying a Signature 87
3.8.3 C# — Complete Sign and Verify Example 88
3.8.4 Higher-Level Libraries 88
3.9 Error Codes 89
Response Format 89
Example Responses 89
3.10 Replay Attack Protection 90
3.11 Passthrough Headers 90
3.12 Route Tracking 90
3.13 Outbound Conflict Resolution 90
3.14 Testing Tool 91
4. Message API (Letterbox) 92
4.1 Overview 92
4.2 API Specification 92
URI Elements 92
4.3 Message Structure 92
Envelope Example 92
Envelope Elements 93
Audit Data Conventions 94
© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

4.4 Successful Response 94
Validation Performed Before 202 94
4.4.1 Per-Message Source-MAP Verification (Step 8 detailed) 95
4.4.2 HMAP-Forwarded Traffic 95
4.5 Synchronous Error Responses 96
HTTP Error Codes 96
Validation Error Codes (9xxx Series) 96
PKI Error Codes (8xxx Series) 97
Synchronous Error Codes (9xxx Series) 97
Error Response Format 97
4.6 Asynchronous Error Handling — Message Delivery Failures 98
Delivery Failure Fields 98
Delivery Failure Codes (9xxx Series) 98
4.7 Message Delivery Policy 99
4.7.1 Architectural framing — synchronous P2P vs hub boundary 99
4.7.2 Required behaviours (CSF MAP-to-MAP) 100
4.7.3 Operational-engagement escalation 100
4.7.4 The HMAP boundary 101
4.7.5 Why exponential backoff (and not linear / fixed-cadence retry) 101
4.7.6 Summary of bindings 102
5. CP Registry API 104
5.1 Overview 104
5.1.1 Side-by-side — TOTSCo vs CSF registry shape 104
5.1.2 Dual-process filtering — same URL pattern across both networks 105
5.2 API Specification 105
5.3 Registry JSON Structure 106
Complete Example 106
MAP in TEST Mode — Onboarding Example 107
5.4 MAP Section — Field Reference 108
5.4.1 Routing Groups 111
5.4.2 Sponsorship Object — Rotation Rules 112
5.5 Directory Section — Field Reference 113
5.6 CP Contact Object 114
Purpose 114
Usage Rules 114
Relationship to MAP Contact Object 114
© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Example Use Case 115
5.7 TOTSCo Directory API 115
6. TOTSCo Integration 116
6.1 Overview 116
6.2 Integration Principles 116
6.3 Registering CSF Endpoints 116
6.3.1 Dedicated, locked-down HMAP interfaces — both directions 117
6.4 Directory Collection 118
6.5 RCPID Format Handling 118
6.6 API Compatibility 118
6.7 Message Flow: CSF to TOTSCo 119
6.8 Message Flow: TOTSCo to CSF 120
6.9 PKI Considerations 120
6.10 Passthrough Headers 120
6.11 Support and Diagnostics 120
6.12 Further Information 121
7. CP Transitions 122
7.1 Overview 122
7.2 Reasons for Moving 122
7.3 Transition Process 122
Step 1 — Establish with New MAP 122
Step 2 — DNS Update 122
Step 3 — Directory Update 123
Step 4 — RCPID Status Export Request 123
Step 5 — In-Flight Order Reconciliation 124
Step 6 — Resume Operations 126
7.4 RCPID Status Export Schema 128
7.4.1 Top-Level Envelope 128
7.4.2 Export Metadata 128
7.4.3 Per-Order Record 128
7.4.4 Worked Example 130
7.4.5 Rules 131
7.5 Conflict Resolution 131
Latency During Transition 131
7.6 RACI for In-Flight Order Handling 132
8. Commercial Scenarios 133

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

8.1 Overview 133
8.2 CP Emergency Migration 133
8.2.1 Definition and Scope 133
8.2.2 Triggers 133
8.2.3 Happy Path — Distressed MAP Still Online 134
8.2.4 Unhappy Path — Distressed MAP Offline 134
8.2.5 Per-Scenario Timing Targets 135
8.2.6 Normal Transition vs Emergency Migration — Contrast 135
8.2.7 Emergency RACI 135
8.2.8 Fee Guidance 136
8.2.9 Audit and Retention 136
8.2.10 CP Emergency Migration Checklist 137
8.3 CP Merger, Acquisition, and Consolidation 137
Stage 1 — Stop Selling 137
Stage 2 — Asset Migration 138
Stage 3 — Close In-Flight Orders 138
Stage 4 — Remove Brands 138
8.4 CP Going into Administration 138
MAP Responsibilities 138
Industry Considerations 139
8.5 Summary of Built-in Protections 139
9. Operations 140
9.1 Monitoring 140
9.2 Malicious Behaviour Detection 140
9.2.1 Directory Hijacking 140
9.2.2 Fraudulent CP Advertisement 140
9.2.3 Traffic Blackholing 140
9.2.4 Denial of Service 140
9.2.5 False Contact Information 141
9.3 Standard Reporting and KPIs 141
9.3.1 Overview 141
9.3.2 Regulatory Reporting 141
9.3.3 Industry Reporting 142
9.4 Outage Notification 142
9.4.1 Design Principles 142
9.4.2 Incident Priority 143
© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

9.4.3 Outage Types 143
9.4.4 JSON Structure 144
9.4.5 Field Reference 145
9.4.6 Receiving MAP Behaviour 146
9.4.7 MAP Obligations 147
9.4.8 Relationship to map.status and Monitoring 148
9.4.9 Feature Advertising 148
9.5 Risk Register 148
9.6 The Right to Block 149
9.6.1 Principles 149
9.6.2 Grounds for Blocking 149
9.6.3 Blocking Procedure 150
9.6.4 Escalation Path 151
9.6.5 Reinstatement 151
9.6.6 Misuse of the Right to Block 152
9.6.7 MoU and Governance Integration 152
Appendix: OpenAPI Specification 153
Overview 153
Specification 153
Usage Notes 161
Content-Type 161
TLS Requirement 161
DKIM Headers 161
Route Tracking 162
Dynamic Message Body Key 162
Error Response Formats 162
© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

SECTION 01 · CSF

Connected Services Framework (CSF)

An Open Transport Framework for the UK Telecommunications Industry

Version 2.0 | Published by the Telecom Technical Architecture Group (TAG)

What is the CSF?

The Connected Services Framework (CSF) is a free, open, and decentralised messaging framework that enables UK Communications Providers (CPs) to exchange messages securely and efficiently through Managed Access Providers (MAPs) — without reliance on a centralised hub or administrative body.

Built on established internet standards (TLS 1.3, OAuth 2.0, DKIM/PKI via RFC 6376) and the industry-standard JSON Asynchronous Messaging Specification (JAM), the CSF provides a peer-to-peer transport layer that any CP or MAP can adopt at zero licensing cost.

Why CSF?

CAPABILITY	CENTRALISED HUB MODEL	CSF (DISTRIBUTED MODEL)
Cost	Ongoing per-message or subscription fees	Free to operate — no licensing or transaction charges
Independence	All parties depend on a single operator	Each MAP operates autonomously
Speed of change	Change requests go through a single vendor	TAG consensus enables rapid iteration
Resilience	Single point of failure	Distributed — no single point of failure
Scalability	Hub capacity limits all participants	Each MAP scales independently
CP freedom	CPs may be locked to a provider	CPs control their MAP choice via DNS
Security	Trust delegated to hub operator	End-to-end PKI signing verifiable by any party
Onboarding	Central registration and approval	Peer-sponsored onboarding — operational in hours
Market entry	Barriers to becoming a provider	Any CP can become a MAP of 1

Initial Use Case: Switching for Business (SforB)

The CSF's first application is delivering **Switching for Business** (SforB), formerly known as Gaining Provider Led Business Switching (GPLB). Unlike One Touch Switch (OTS) for residential customers, Ofcom has not mandated a single technical solution for business switching — enabling the industry to adopt this open, distributed approach.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

The framework is not limited to SforB. It is designed to support any industry messaging process, including future applications such as number porting and bilateral communications.

Documentation Structure

Reference

DOCUMENT	DESCRIPTION
Definitions & Terminology	Consolidated reference for all acronyms, abbreviations, and CSF-specific terminology
CP Rights Charter	Plain-language summary of CP rights under the CSF — standalone, no technical background required
MAP Onboarding & Testing Process	Practical handbook for developers, implementers, and testers — environments, status model, NowYoYo PKI test harness, and the TAG Baseline Test Suite

This documentation is organised into two parts:

Part 1 — Framework (Management & Architecture)

For decision-makers, solution architects, and programme managers.

DOCUMENT	DESCRIPTION
Introduction	Purpose, scope, and value proposition of the CSF
Background	OTS history, SforB context, and regulatory landscape
Principles & Requirements	Gathered requirements with MoSCoW prioritisation
Architecture Overview	Layered model, message flow, and entity definitions
CP Registry, Master Registry & Directory	Three-tier distributed registry model and synchronisation
Security Overview	TLS 1.3, OAuth 2.0, and PKI/DKIM at a conceptual level
Onboarding	MAP sponsorship and CP registration processes
RCPID Management	UUIDv4 identifiers — allocation, transitions, and rationale
Operational Excellence	Resilience, SLAs, monitoring, and deployment patterns
Governance	Standards bodies, MoU, and change control
Glossary	Abbreviations and definitions

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Part 2 — Implementation (Technical)

For developers, integration engineers, and technical architects.

DOCUMENT	DESCRIPTION
Getting Started	Prerequisites and onboarding checklist
OAuth 2.0 Implementation	Token requests, responses, and error handling
PKI & DKIM Signing	DKIM process, CSF headers, worked examples, and error codes
Message API	Letterbox API, HTTP codes, message delivery failures, and the `auditData` conventions for test flagging and match-session sequencing
Directory API	Registry JSON schema, MAP-level (`map.status`) and per-CP (`processSupport[].status`) status fields, and field reference
TOTSCo Integration	Hub interoperability and dual-endpoint configuration
CP Transitions	Six-step migration process, RCPID Status export schema, suggested in-flight order export routine, Step 6 sign-off checklist, and conflict resolution
Commercial Scenarios	MAP failure, CP mergers, and administration handling
Operations	Monitoring, malicious behaviour, regulatory and industry reporting (with the proposed distributed telemetry collection model), and risk register
FAQ — GPLB Steering Group	GPLB-SG questions and answers
FAQ — TOTSCo Integration	TOTSCo integration questions and answers
Appendix: OpenAPI Specification	Full OpenAPI 3.0.3 spec for the Letterbox API

Standards & External References

The CSF builds on and references the following industry standards:

- **JSON Asynchronous Messaging Specification (JAM)** — published on the OTA2 website
- **RCPID Standards** — published on the OTA2 website
- **[TOTSCo Hub API Specification v2.0](https://totsco.org.uk/wp-content/uploads/2026/04/TOTSCo-API-specifications-v2-Clean.pdf)** — the industry-standard letterbox and directory API for SforB, used by TOTSCo and supported by CSF for hub interoperability
- **Switching for Business Process Documents** — published on the FCS website
- **GPLB-SG Switching for Business Message Specification v3.0** — bundled with this repo as `framework/GPLB SG Switching for Business Message Specification V3.0.pdf`; the authoritative low-level SforB message spec (canonical source published by the GPLB-SG via FCS)
- **TOTSCo Process & Technical Documents** — published on the TOTSCo website

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Key RFCs

RFC	USAGE IN CSF
RFC 8446	TLS 1.3 — mandatory transport encryption
RFC 6376	DKIM — message signing and verification
RFC 6749	OAuth 2.0 — authorisation framework
RFC 4122	UUIDv4 — RCPID format
RFC 8463	Ed25519 for DKIM — alternative signing algorithm

Reading the Working Draft

The **working draft** of the CSF is held by the **Telecom Technical Architecture Group (TAG)**. The TAG manages change control, version increments, and the editorial lifecycle of the document set.

Published versions are made available for download from the CSF website. Each published version is signed off by the TAG and superseded only by a subsequent published version through the TAG's change-control process.

Contributing

The CSF is maintained by the Telecom Technical Architecture Group (TAG). The TAG steering group meets weekly to review changes and proposals. All CSF documentation is under TAG change control.

Contributors

NAME	ORGANISATION
Dave Stubbs	Zentive Limited
David A Wilson	iconectiv
Marcel Horst	CWP
Mark Delo	NowYoYo Limited
Shanmugavel Krishnan	iconectiv
Simon Phillips	ICUK Computing Services Limited
Stephen Breen	iconectiv
Steven Pearce	NowYoYo Limited
Susan Mangini	iconectiv

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Version History

VERSION	DATE	DESCRIPTION
2.0.1	2026-04-29	Content additions: new MAP Onboarding & Testing Process handbook with the TAG Baseline Test Suite; expanded CP Transitions §7.3–§7.4 with detailed step descriptions, suggested in-flight order export routine, full RCPID Status export schema, and Step 6 sign-off checklist; new optional `map.status` enum (`ACTIVE`/`TEST`/`SUSPEND`) in the Directory API; tightened telemetry rules in §10.6 (collection limited to `ACTIVE` participants, `auditData.test=true` exclusion); new auditData conventions in the Letterbox doc cross-linking the test-flag and match-session-sequencing rules; bundled GPLB-SG SforB Message Specification v3.0 PDF as the authoritative low-level reference.
2.0	2026-04-24	Comprehensive rewrite in Markdown. Restructured into separate Part 1 (Framework) and Part 2 (Implementation) documents with deduplication, consistent terminology, and improved technical accuracy.
1.0.5	2026-01-07	Last Word document version (Part 1 — Framework)
1.0.5	2025-11-14	Last Word document version (Part 2 — Implementation)

Licence

The Connected Services Framework is an open transport framework, free for any Communications Provider or Managed Access Provider to use for any purpose. There are no licensing fees or transaction charges. Participation is governed by the Memorandum of Understanding (MoU) agreed among participating MAPs and CPs.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

PART 1

Framework

Management & Architecture — for decision-makers, solution architects, and programme managers

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

1. Introduction

1.1 Purpose of This Document

This document defines the Connected Services Framework (CSF), an open transport framework for the UK telecommunications industry. It establishes the technical and operational prerequisites, architectural design, and implementation strategy that enable Managed Access Providers (MAPs) to exchange messages securely and efficiently on behalf of their Communications Providers (CPs).

The CSF is free for any organisation to adopt. There are no licensing fees, transaction charges, or commercial dependencies on any single provider.

1.2 What the CSF Does

The CSF is a **transport layer** — it defines how messages are wrapped, signed, and delivered between MAPs. It does not define the content of those messages. Industry processes such as Switching for Business (SforB) or One Touch Switch (OTS) define the message content; the CSF delivers it.

In practical terms, the CSF provides:

- **Secure peer-to-peer message delivery** between MAPs using TLS 1.3, OAuth 2.0, and DKIM-based PKI signing
- **A distributed registry model** where each MAP publishes a CP Registry of its CPs, all MAPs converge these into a private Master Registry, and each MAP derives a filtered Directory for its own CPs
- **DNS-based CP verification** that puts CPs in control of which MAP represents them
- **Standardised onboarding** through peer sponsorship, removing the need for centralised administration
- **In-flight order portability** so CPs can change MAPs without disrupting active switching orders

1.3 Who This Document Is For

Part 1 (this document) is written for:

- **Senior management and decision-makers** evaluating whether to adopt the CSF
- **Solution architects** designing CP or MAP implementations
- **Programme managers** planning SforB delivery

Part 2 (Implementation) is written for developers and integration engineers who need API specifications, code examples, and JSON schemas.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

1.4 Scope

The CSF covers:

- The definition of a decentralised messaging architecture independent of any centralised administrative authority
- Requirements for interoperability, security protocols, and communication standards between MAPs
- Protocols for onboarding new MAPs and transitioning CPs between MAPs
- Guidelines for maintaining data integrity, ensuring secure communications, and managing exceptional scenarios such as MAP failures, CP migrations, mergers, and administration

The CSF does **not** cover:

- How a MAP integrates with its own CPs (this is each MAP's commercial decision)
- The content of industry process messages (defined by OTS, SforB, etc.)
- The internal architecture of any MAP's platform

1.5 The Case for a Distributed Model

The UK telecoms switching landscape currently operates through a centralised hub model for residential switching (OTS via TOTSCo). While this model served its purpose, it introduces constraints that are unnecessary for business switching:

Cost: A centralised hub requires ongoing funding — through per-message fees, subscriptions, or industry levies. The CSF eliminates these costs entirely. MAPs exchange messages directly with each other at no charge.

Independence: In a hub model, every CP and MAP depends on a single operator for message routing, change management, and availability. The CSF distributes these responsibilities across all participants.

Speed of innovation: Hub changes require approval from a single vendor's change control process. CSF changes are agreed by the TAG steering group and can be adopted incrementally by MAPs through versioning and feature releases.

Resilience: A centralised hub is a single point of failure. The CSF's peer-to-peer architecture means that an outage at one MAP affects only that MAP's CPs — not the entire industry.

CP freedom: The CSF gives CPs direct control over their MAP relationship through DNS. A CP can change MAP by updating two DNS records, and their RCPID stays with them for life.

Market access: Any CP can become a MAP of 1 (serving only itself) and exchange messages directly with the industry. This lowers barriers to entry and avoids monopolistic lock-in.

Additional routes, not a replacement. MAPs and CPs can already exchange SforB traffic today via the TOTSCo Hub. The CSF does not replace that route — it adds **two additional, complementary routes** for the industry:

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- **Managed via a CSF MAP** — a CP subscribes to a CSF MAP and exchanges through that MAP's CSF outbound. The CSF supports multiple competing MAPs, giving CPs real commercial choice.
- **Self-served as a MAP-of-1** — a CP runs its own MAP infrastructure, exchanging messages directly with every other MAP on the CSF network.

Together with the existing TOTSCo Hub route, this gives the industry three concurrent routes. The deliberate intent is **rapid SforB expansion**: many smaller business CPs need to become compliant in a short window, and offering them more than one route — without losing the TOTSCo Hub option — is the lever that makes industry-wide adoption practical.

> NOTE
> : The CSF does not preclude the use of a centralised hub. TOTSCo can operate as a Hub MAP (HMAP) within the CSF, allowing CPs connected to the hub to exchange messages with CPs connected to other MAPs seamlessly. The HMAP interface each MAP exposes to TOTSCo is **dedicated to TOTSCo and locked down** — CSF spec increments do not propagate through it, so TOTSCo's integration is a one-time build, not an evergreen one. See Standalone — TOTSCo HMAP Integration Guide for the full operating model.

> CSF DIRECTORY IS TOTSCO-COMPATIBLE BY DESIGN.
> The CSF CP Registry (Part 2 §5) was designed as a **strict superset** of the TOTSCo OTS/GPLB directory shape. Parties that already consume the TOTSCo directory can ingest the CSF Registry with their existing parser; CSF-specific additions are additive and never breaking. This was a deliberate design decision so the industry could operate easily with the CSF through the JSON contract alone.

1.6 Initial Application: Switching for Business (SforB)

The CSF's first industry application is **Switching for Business (SforB)**, formerly known as Gaining Provider Led Business Switching (GPLB).

Key differences from OTS:

- **No mandated hub**: Ofcom has not required a centralised technical solution for business switching
- **Gaining-provider-led**: The switching process is initiated by the gaining provider
- **Larger scale**: The industry expects a tenfold increase in RCPIDs compared to OTS
- **Longer onboarding**: Industry-wide ramp-on is expected to span a considerably longer timeframe

The CSF has been designed to support SforB as its initial use case, but is architected for any future industry messaging process — including number porting, bilateral communications, and processes not yet defined.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

1.7 Document Structure

CHAPTER	CONTENT
Background	History of OTS, the emergence of SforB, and the regulatory context
Principles & Requirements	Industry-gathered requirements with MoSCoW prioritisation
Architecture Overview	Layered model, message flow, and entity definitions
CP Registry, Master Registry & Directory	The three-tier distributed registry model
Security Overview	Multi-layered security architecture
Onboarding	MAP sponsorship and CP registration
RCPID Management	Identifier format, allocation, and lifecycle
Operational Excellence	Resilience, SLAs, and deployment patterns
Governance	Standards bodies and change control
Glossary	Terms and abbreviations

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

2. Background

2.1 The Evolution of UK Telecoms Switching

The United Kingdom's telecommunications sector has undergone significant evolution, driven by government policy objectives aimed at promoting a more competitive, consumer-focused market. Central to this transformation is the **Switching General Conditions**, a regulatory framework designed to streamline switching processes by minimising inefficiencies and eliminating procedural redundancies.

2.2 One Touch Switch (OTS): Establishing the Foundation

Through the Switching General Conditions, the industry formed a Detail Design Group (DDG) to establish the requirements for switching residential customers' broadband and voice services — what Ofcom termed "One Touch Switch."

The DDG established several foundational principles that the CSF builds upon:

1. **JSON message format**: A standardised format separating delivery instructions (the envelope) from the message content (the body)
2. **Shared directory**: A directory of all participating CPs that could be shared by all participants
3. **Common connectivity**: A standardised way for CPs to connect, send, and receive messages
4. **Multiple delivery platforms**: Support for multiple solutions — hub, portal, and direct integration — provided by what are now known as Managed Access Providers (MAPs)

These principles now form the basis of the **JSON Asynchronous Messaging Specification (JAM)**, published on the OTA2 website and used by both OTS and SforB.

Limitations of the OTS Implementation

To guarantee timely delivery of OTS, certain original design principles were not fully implemented:

- Ofcom mandated that all OTS messages be routed through a single centralised hub (TOTSCo), limiting direct MAP-to-MAP exchange
- Industry processes for CP transitions between MAPs were not addressed
- RCPID management — where the CP, not the MAP, owns the identity — was not fully resolved
- The directory model, while distributed in concept, remained dependent on a central operator

These gaps created the need for the TAG and the CSF.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

2.3 Switching for Business (SforB): Removing Restrictions

Mandated by Ofcom, the business switching process was established on principles similar to OTS, with one critical distinction: **Ofcom has not mandated that all messages be routed through a centralised hub**.

This removal of the routing restriction enables MAPs and individual CPs to exchange messages directly with one another, rather than through a central intermediary.

The Switching for Business process:

- Establishes the **gaining-provider-led model** as mandated by Ofcom
- Adopts the **Switching for Business (SforB) process** — stewarded by the GPLB Steering Group (GPLB-SG) — uniformly across the industry
- **Decouples** CPs from the constraints of any specific transport mechanism
- Represents the **first industry process** designed to streamline direct inter-provider messaging

2.4 The Telecom Technical Architecture Group (TAG)

The TAG is a consortium of telecom industry stakeholders — including MAPs, System Integrators, and CPs — convened to:

1. Identify gaps in the OTS messaging infrastructure
2. Establish core requirements for extending the OTS foundation to support SforB and future processes
3. Design and maintain the Connected Services Framework

The TAG collaborates closely with the GPLB Steering Group (GPLB-SG) and references the Switching for Business process documentation published on the FCS website.

Before designing any solution, core requirements were gathered from all MAPs, System Integrators, and related CPs. These requirements — covering cost reduction, security enhancement, and message exchange processing — form the foundation of the CSF as documented in Principles & Requirements.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

2.5 From OTS to CSF: What Changed

ASPECT	OTS	CSF
Routing	All messages via TOTSCo Hub	Direct MAP-to-MAP (peer-to-peer)
Hub dependency	Mandatory	Optional (TOTSCo can operate as HMAP)
Directory	Centralised at TOTSCo	Each MAP publishes its own registry
RCPID format	4-character codes (e.g., RGXD)	UUIDv4 for scalability and uniqueness
CP identity	Managed by MAP/Hub	CP controls via DNS records
Message signing	Hub validates	DKIM-based PKI — any recipient can verify
Onboarding	Central registration	Peer-sponsored, self-service
Cost model	Hub subscription/per-message fees	Free — no charges between MAPs
CP portability	Limited	Built-in: RCPID follows the CP for life
Change control	Hub vendor process	TAG steering group (weekly)

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

3. Principles and Requirements

3.1 How Requirements Were Gathered

The requirements documented in this chapter were explicitly gathered from Managed Access Providers (MAPs), Systems Integrators (SIs), and their associated CPs. The purpose was to define how MAPs communicate with each other efficiently, reliably, and securely to facilitate messaging solutions on behalf of their CPs for any industry process.

The intention is that modifications stipulated within the CSF will not impact solutions already developed by CPs in support of OTS. However, MAPs may need to implement changes to fully support distributed messaging for processes such as SforB.

3.2 Requirements

Requirements are prioritised using the MoSCoW method:

- **Must** — Non-negotiable; the CSF cannot function without this
- **Should** — Important but not blocking; deviation requires justification
- **Could** — Desirable if achievable without significant cost or complexity
- **Won't** — Out of scope for the current version

Connectivity and Routing

REF	REQUIREMENT	PRIORITY
R.01	MAPs MUST be free to acquire and manage their own customers without a technical or commercial dependency on, or imposed by, another MAP. No centralised administration is required.	Must
R.02	MAPs MUST make their customers available for message exchange with any other requesting MAP without applying a commercial or technical barrier.	Must
R.03	Unless required by regulation or an agreed industry process, no single party SHALL be a dependency for routing messages to any other MAP and their CPs.	Must
R.04	All OTS messaging MUST continue to be routed via TOTSCo to meet Ofcom compliance requirements.	Must
R.05	Direct routing MUST NOT be mandated between MAPs. A MAP can choose who they route through unless required by regulation or an agreed industry process.	Must
R.06	MAPs MUST ensure they have peering capabilities to reach all CPs in the directory.	Must
R.07	Any MAP MAY choose to provide routing services to another MAP (e.g. TOTSCo), but MUST NOT route to another routing MAP as circular routes MUST be avoided.	Must
R.08	MAPs MUST meet the requirements of the industry processes in delivering messages and meeting CP SLAs.	Must
R.09	SLAs MUST be defined for all MAP-to-MAP transactions.	Must
R.10	MAPs MUST maintain a secure networking environment between all MAPs and their CPs.	Must

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

REF	REQUIREMENT	PRIORITY
R.11	MAPs MUST ensure they only accept messages from authenticated CPs and MAPs.	Must
R.12	Every MAP MUST provide the same guaranteed delivery capability to its CPs, including retry mechanisms and failure reporting.	Must
R.13	A mechanism is REQUIRED to allow auto-discovery of CPs by onboarded MAPs.	Must
R.14	The solution MUST support message sending and receiving in compliance with supported industry processes (e.g., OTS, SforB).	Must
R.15	The solution MUST be as simple as possible to implement.	Must

Directory Management

REF	REQUIREMENT	PRIORITY
R.17	A centralised directory SHOULD NOT be a dependency or requirement.	Should
R.18	Every MAP MUST provide its customers with access to a complete directory list in support of their messaging processes.	Must
R.19	Every MAP is responsible for maintaining and publishing the CP Registry entries of its own CPs.	Must
R.20	All MAPs MUST be able to issue their own unique RCPIDs.	Must
R.21	A new MAP MUST have a sponsor and be onboarded before being added to the network.	Must
R.22	Any shared directory model MUST be openly extensible.	Must

Messaging

REF	REQUIREMENT	PRIORITY
R.23	Messages SHOULD be capable of being encrypted end-to-end.	Could
R.23a	All messages exchanged between MAPs MUST be digitally signed using DKIM (RFC 6376) so that the receiving MAP can verify the message has not been tampered with in transit and that it originated from the claimed source CP.	Must

Operations

REF	REQUIREMENT	PRIORITY
R.24	No MAP MAY charge for connectivity and routing of messages from or to another MAP, or levy charges to another MAP's customers.	Must
R.25	Every MAP MUST ensure they have connectivity to exchange and route messages.	Must

Other

REF	REQUIREMENT	PRIORITY
R.26	Linking RCPIDs (e.g., association of OTS and SforB RCPID if a CP uses one for each process).	Could

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

CP Right to Move

REF	REQUIREMENT	PRIORITY
R.27	Every CP MUST have the right to change to another MAP.	Must
R.28	No MAP SHALL actively prevent a CP from moving to a new MAP.	Must
R.29	No MAP SHALL deliberately prevent access to switching for a CP while still under contract when the CP is switching to another MAP.	Must
R.30	A CP MUST be able to choose their service status within the MAP's directory, regardless of account status (e.g., ACTIVE, TEST, SUSPENDED).	Must

3.3 Summary of Principles

Taken together, these requirements establish that the CSF must:

1. **Ensure MAP independence** — No commercial or technical dependency on any other MAP
2. **Guarantee interoperability** — Secure routing of messages between all MAPs
3. **Remove barriers** — MAPs provide connectivity without imposing technical or commercial barriers
4. **Maintain security** — Authenticated message exchanges using OAuth 2.0, PKI, and DNS-based validation
5. **Enable auto-discovery** — Decentralised discovery of MAPs and CPs through shared directory structures
6. **Guarantee delivery** — Consistent message delivery with retry mechanisms and failure reporting
7. **Empower CPs** — Freedom and autonomy to switch MAPs without disruption or obstruction
8. **Enforce SLAs** — MAP-to-MAP service-level agreements for consistency and reliability
9. **Comply with regulation** — All MAP activities must comply with relevant industry regulations
10. **Reward good behaviour** — The framework encourages good actors and discourages bad actors without requiring centralised enforcement

These principles support operational excellence, regulatory compliance, and industry-wide interoperability, fostering an environment conducive to innovation and dynamic market competition.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

4. Architecture Overview

4.1 Layered Architecture

The CSF operates as a layered messaging stack. Each layer has a distinct responsibility, and together they provide a complete, secure message delivery pipeline.

LAYER	ROLE	STANDARDS / MECHANISMS
Process Layer	Defines the business message content (e.g., SforB Switch Match, Switch Order)	Open industry process message formats (defined by GPLB-SG, OTS-DDG, etc.)
JAM Routing Layer	Defines addressing, message correlation, and routing	JSON Asynchronous Messaging Specification (RCPID addressing, envelope headers)
CSF Security Layer	Handles message signing, encapsulation, and verification	PKI/DKIM (RFC 6376), OAuth 2.0, DNS-based validation
Transport Layer	Ensures secure delivery over the network	HTTPS over TLS 1.3 (RFC 8446)

The process layer and JAM routing layer are industry-wide standards that exist independently of the CSF. The CSF's contribution is the security layer and the operational framework that binds everything together.

4.2 Entity Definitions

4.2.1 Communications Provider (CP)

A CP is any party participating in an industry process — creating or consuming messages and sending or receiving them from other CPs. CPs include retailers, wholesalers, and agencies providing services on behalf of other CPs.

The CP involved will vary based on the industry process. A CP does not need to understand how the CSF works — it creates messages according to the industry process specification and hands them to its MAP for delivery.

4.2.2 Managed Access Provider (MAP)

A MAP facilitates message exchange on behalf of one or more CPs. MAPs provide integration services, portals, technical solutions, and the operational infrastructure to send and receive messages over the CSF.

Within the CSF, a MAP is the **only entity** authorised to exchange messages with other MAPs, because MAPs are responsible for:

- Publishing and maintaining their CP registry
- Applying PKI signing to outbound messages
- Verifying PKI signatures on inbound messages
- Establishing and maintaining OAuth 2.0 credentials with other MAPs

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- Guaranteeing message delivery with retry mechanisms

A MAP may serve multiple roles concurrently for its clients (insourced, technical, fully managed), and the CSF imposes no restrictions on this. There are no charges or licensing requirements mandating specific MAP definitions.

4.2.3 MAP of 1

Any CP can become a "MAP of 1" — operating as its own MAP. This allows CPs to exchange messages directly with the industry for free, maintaining full control of all messages sent and received from their own infrastructure. The CP must abide by all MAP rules when implementing.

4.2.4 Hub Managed Access Provider (HMAP)

An HMAP is a specialised MAP designed to facilitate integration for proxy hubs such as TOTSCo. An HMAP:

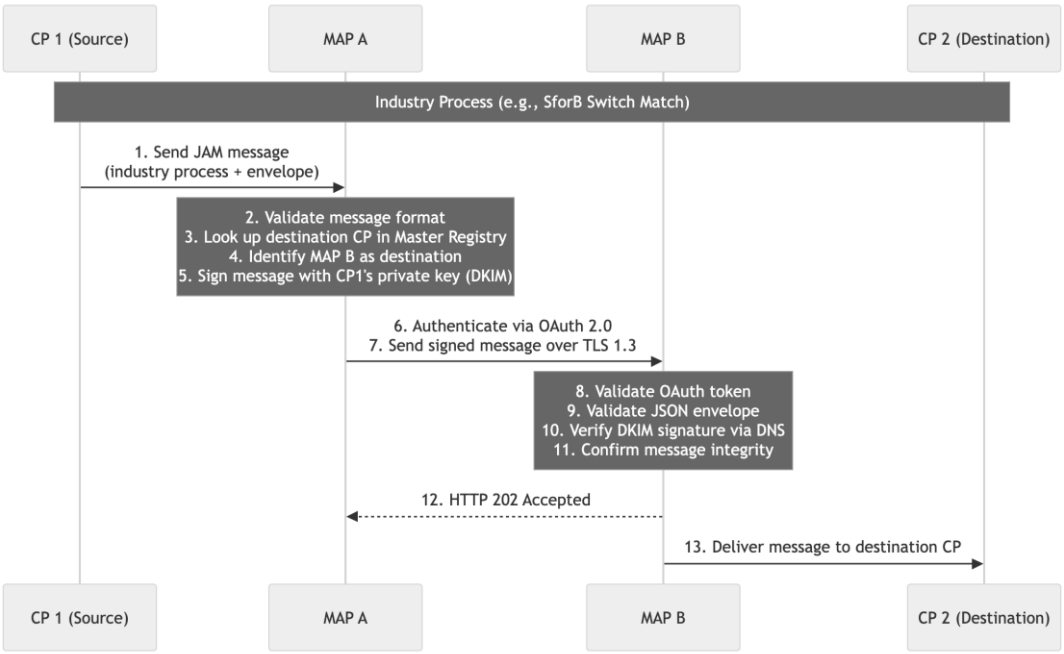
- Accepts messages and forwards them among its subscribed members using a hub-and-spoke model
- Does **not** engage with the message payload — it routes based on addressing information
- When interfacing with the CSF, uses its Master Registry (built from collected CP Registries) to locate the destination CP and dispatches the message to the appropriate MAP
- Is **not required** to implement PKI/DKIM signing when exchanging messages on behalf of its CPs (to minimise integration cost)
- **Is strongly recommended** to implement the full CSF CP Registry format for SforB, as its CP clients benefit significantly from the additional information it provides

> ✓ BEST PRACTICE
> : MAPs on the CSF should provide a dedicated endpoint exclusively for HMAPs, separate from the endpoint used by other CSF MAPs. This allows rapid CSF iteration without disrupting established hub exchange networks.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

4.3 Message Flow

The following diagram illustrates the end-to-end message flow through the CSF:



Step-by-Step Workflow

1. **CP creates message**: The CP creates a message according to the industry process specification (e.g., SforB) and wraps it in a JAM envelope with RCPID addressing, using directory information provided by its MAP.
2. **CP sends to MAP**: The CP sends the message to its MAP using whatever integration method the MAP provides (API, portal, BSS integration). MAP-to-CP integration is outside the scope of the CSF.
3. **MAP validates**: The MAP validates the message structure, format, and addressing. Unlike a hub model, MAPs **can** read the full CP payload and validate compliance before signing.
4. **MAP discovers route**: Using its Master Registry, the MAP identifies which MAP serves the destination CP.
5. **MAP signs message**: The MAP signs the message with the source CP's private key using DKIM (RFC 6376). The signature is placed in HTTP headers — the message body is not altered.
6. **MAP authenticates**: The sending MAP authenticates with the receiving MAP via OAuth 2.0.
7. **MAP delivers**: The signed message is sent over TLS 1.3 to the receiving MAP's letterbox endpoint.
8. **Receiving MAP verifies**: The receiving MAP validates the OAuth token, checks the JSON envelope, retrieves the source CP's public key from DNS, and verifies the DKIM signature to confirm the message is authentic and unaltered.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

9. **Receiving MAP acknowledges**: On successful validation, the receiving MAP returns HTTP 202 (Accepted) — indicating the message has been accepted for processing but processing has not yet completed.
10. **Receiving MAP delivers**: The receiving MAP delivers the message to the destination CP.

> NOTE
> : If both CPs use the same MAP, the CSF is typically not needed as there is no inter-MAP exchange. However, MAPs operating multiple instances or solutions may elect to use the CSF for their own internal interconnections.

4.4 CSF Networks

Message exchanges are conducted solely between MAPs. The CSF may be employed across diverse industry processes, and the network adapts its size to the specific process in use.

Each MAP must:

1. **Undergo sponsored onboarding** with an existing MAP that supports the relevant industry process
2. **Establish secure connectivity** with all other MAPs (OAuth 2.0 credentials, endpoint registration)
3. **Publish its CP Registry** detailing its CPs and supported processes
4. **Retrieve all other MAPs' CP Registries** to build its Master Registry

Feature-based services are advertised in the CP Registry, allowing certain MAPs to utilise additional functionalities for bilateral and advanced communication. This design ensures MAPs transmit only messages that the receiving MAP can process.

4.5 API Versioning and Feature Releases

The CSF supports incremental evolution through two mechanisms:

API Versioning

MAPs can support multiple API versions simultaneously, allowing new features to be adopted without disrupting existing integrations. When both parties support the latest version, they automatically use advanced features. Where compatibility varies, messages default to the highest compatible version.

Feature Discovery

A MAP queries the directory to identify the supported feature set of another MAP. This enables:

- **Backward compatibility**: All MAPs maintain compatibility with earlier API versions

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- **Flexible adoption**: MAPs can adopt new features at their own pace
- **Rapid iteration**: The TAG can release new capabilities without requiring industry-wide synchronisation

By implementing versioning and feature discovery, the CSF enables rapid feature adoption while reducing the constraints of traditional messaging protocols.

4.6 What the CSF Does Not Prescribe

The CSF is deliberately silent on:

- **MAP-to-CP integration**: How a MAP connects to its CPs is a commercial decision (API, portal, BSS plugin, etc.). Typically, MAPs will attempt to offload or shield CPs from as much of the heavy lifting for a process like SforB as possible, and this is often used to differentiate between MAPs, but this is a technical-commercial decision outside the scope of this framework.
- **Internal MAP architecture**: MAPs are free to use any technology stack
- **CP message creation**: CPs create messages according to industry process specifications, not CSF specifications
- **Commercial arrangements**: MAPs and CPs establish their own commercial relationships independently of the CSF

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

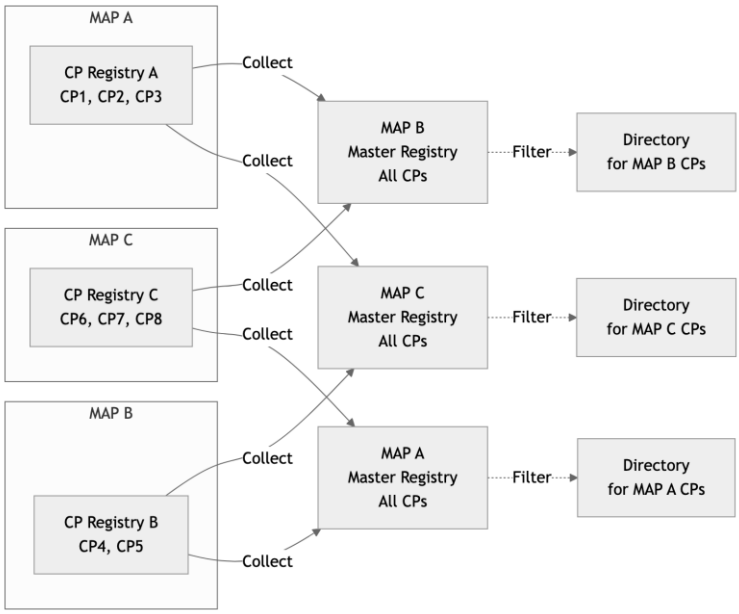
5. CP Registry, Master Registry, and Directory

5.1 Overview

The distributed registry model is a foundational pillar of the CSF. Unlike a centralised model where a single operator maintains the master list of all CPs, the CSF distributes this responsibility across all MAPs using a three-tier approach.

Each MAP:

1. **Publishes a CP Registry** — a JSON document listing its own CPs and their metadata, made available to other MAPs via an OAuth 2.0-protected API endpoint
2. **Collects all other MAPs' CP Registries** — via those same API endpoints
3. **Converges these into a Master Registry** — a private, consolidated view containing full routing, PKI, and contact details for all CPs across the entire CSF network
4. **Derives a Directory for its CPs** — a filtered, cut-down version of the Master Registry listing just the brand names and RCPIDs available to switch with



How the Three Tiers Work Together

1. A CP selects a destination brand from the **Directory** (e.g., a drop-down list in the MAP's portal)
2. The MAP uses the selected RCPID to look up the full routing and signing details in its private **Master Registry**
3. The Master Registry tells the MAP which MAP serves the destination CP, which endpoint to send to, and which PKI domain to use for verification
4. The destination CP's details originally came from that MAP's **CP Registry**, which was collected and converged into the Master Registry

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

5.2 CP Registry Structure

Each MAP's CP Registry contains two main sections:

MAP Section

Comprehensive metadata about the MAP itself:

- **Identity**: MAP name, version timestamp
- **Contact information**: Phone, email, URL with availability hours and days
- **Connectivity**: CP Registry API endpoint URL, OAuth token URL, messaging service endpoints
- **Routing groups**: Which message types (e.g., `businessSwitch.*`, `residentialSwitch.*`) are accepted at which endpoint
- **Connections**: List of other MAPs this MAP has established connectivity with
- **Registration URI**: Where new CPs or MAPs can apply to register
- **Service status**: URL and API endpoint for checking operational status and planned changes

CP List Section

For each CP registered with the MAP:

- **RCPID**: The CP's unique identifier (UUIDv4)
- **Brand name**: The human-readable name shown in search/drop-down lists
- **Process support**: Which industry processes the CP participates in (OTS, SforB) and their status (ACTIVE, SUSPENDED, TEST)
- **Resources**: Customer assist URLs, sales assist URLs, switch support contacts — backward compatible with OTS resource fields
- **Signing information**: The DNS domain used for PKI key lookup and which routing IDs should be signed with that domain
- **Contact details** (optional): A `contact` array using the same structure as the MAP contact object — phone, email, URL with availability hours, days, and purpose (sales, support, technical). This allows each CP to advertise its own preferred contact methods for operational escalations, independent of the MAP's contacts

> **TIP**
> For the full JSON schema and field reference, see Part 2: Directory API.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

5.3 CP Registry Synchronisation

MAPs retrieve other MAPs' CP Registries via HTTP GET requests to each MAP's registry API endpoint, authenticated with OAuth 2.0 credentials established during onboarding.

Frequency

- There is no mandatory synchronisation interval — MAPs can pull as frequently as needed
- A maximum frequency of approximately **60 seconds** is recommended to avoid unnecessary load
- MAPs MUST retrieve CP Registries regularly and in near-real time, particularly as new CPs are onboarded continuously throughout the day

Building the Master Registry

Each MAP independently converges the collected CP Registries into its own private copy of the Master Registry. The Master Registry contains:

- **Full routing information**: Endpoint URLs, OAuth token URLs, and routing group definitions for every MAP
- **PKI signing domains**: The DNS domains used for DKIM signature verification for every CP
- **Contact details**: MAP and CP contact information for operational support and escalations
- **Service status**: Real-time availability indicators for each MAP

Deriving the Directory

From the Master Registry, each MAP creates a Directory for its own CPs. The Directory contains only the information CPs need to address messages:

- **Brand names** and **RCPIDs** — for drop-down lists and search-as-you-type
- **Supported processes** — which CPs support OTS, SforB, etc.
- **Resource URLs** — customer assist and sales assist URLs (backward compatible with OTS)

The Directory excludes inter-MAP routing details, PKI signing domains, OAuth endpoints, and other sensitive operational information. How the MAP presents the Directory to its CPs (portal, API, BSS integration) is the MAP's commercial decision.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

5.4 Advantages of the Distributed Model

BENEFIT	DESCRIPTION
Data accuracy	Each MAP is the authoritative source for its own CPs, reducing inaccuracies
Real-time updates	New CPs can be onboarded and advertised in seconds, without central approval
Scalability	Each MAP manages only its own CP Registry — the network scales naturally
No single point of failure	If one MAP is offline, all other MAPs continue to operate
Extensibility	The CP Registry schema can be extended without breaking existing implementations
CP control	CPs control their MAP association via DNS, not through a central register, which means CPs stay in control at all times, not the MAP.
Transparency	Every MAP can see the full network state by converging all CP Registries into its Master Registry

5.5 Conflict Resolution

When a CP appears in more than one MAP's CP Registry simultaneously (typically during a CP transition between MAPs), the conflict is resolved using DNS:

1. The MAP performs a DNS lookup on the CP's `_mapkey` record
2. The DNS record indicates the CP's current MAP endpoint
3. The MAP routes messages to the MAP indicated by DNS, regardless of which CP Registry listed the CP
4. The MAP marks the conflicting entry as being in "conflict" in its Master Registry

This DNS-based resolution ensures that the CP — not the MAP — has ultimate control over routing.

> **TIP**
> For detailed conflict resolution procedures, see Part 2: CP Transitions.

5.6 TOTSCo Compatibility

The CSF CP Registry structure is backward compatible with the TOTSCo OTS directory. The same fields used in OTS (RCPID, brand name, customer assist URL, sales assist URL) are preserved in the CSF CP Registry using the same structure.

TOTSCo can collect CSF MAPs' CP Registries and either merge them into its own Master Registry or maintain them separately. The CP Registry is JSON-formatted, making convergence straightforward.

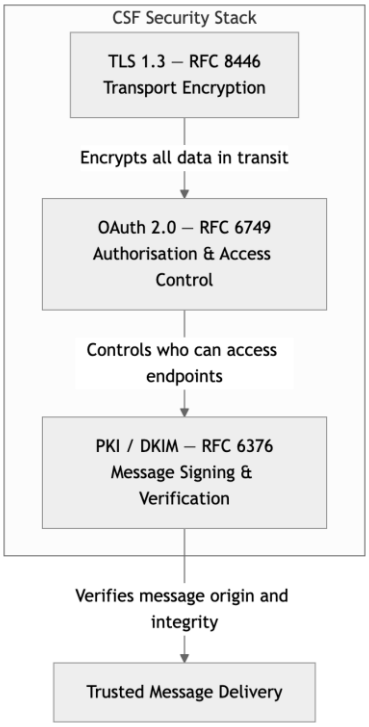
For details on consuming the TOTSCo Directory API, see Part 2: TOTSCo Integration. The TOTSCo Hub API Specification v2.0 defines the standard directory and letterbox interfaces used by TOTSCo and supported by the CSF for hub interoperability.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

6. Security Overview

6.1 Security Architecture

The CSF implements a multi-layered security architecture combining three established standards. Together, they deliver end-to-end encryption, authenticated access, and verifiable trust for every connection, credential, and message.



LAYER	STANDARD	PURPOSE
Transport	TLS 1.3	Encrypts all data in motion, protecting against interception and tampering
Access Control	OAuth 2.0	Ensures only authorised MAPs and CPs can interact with API endpoints
Message Integrity	PKI/DKIM	Digital signatures verify message authenticity and that content has not been altered

This combination delivers a **zero-trust, cryptographically assured trust boundary** across networks and services.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

6.2 Transport Layer Security (TLS 1.3)

TLS is a cryptographic protocol providing secure communication over computer networks. It ensures that data transmitted between MAPs remains confidential, authentic, and integrity-protected.

What TLS 1.3 Provides

- **Encryption**: Protects data from eavesdropping using symmetric encryption after a secure handshake
- **Authentication**: Verifies server identity using digital certificates from trusted Certificate Authorities
- **Integrity**: Ensures data has not been tampered with during transit through message authentication codes

Why TLS 1.3 Specifically

TLS 1.3 (RFC 8446) is mandated by the CSF because it:

- Removes legacy insecure algorithms (RSA key exchange, SHA-1, static DH)
- Reduces handshake latency from two round-trips to one
- Mandates **forward secrecy** — session keys cannot be recovered even if a server's private key is compromised
- Supports only strong, modern encryption (AES-GCM, ChaCha20-Poly1305)
- Encrypts more of the handshake, hiding metadata from passive observers
- Has growing support for Post-Quantum cryptography (NTRU, ML-KEM, etc.)

> ✓ CSF REQUIREMENT
> : TLS 1.3 is mandatory for all CSF deployments. Earlier TLS versions MUST NOT be used.

6.3 OAuth 2.0

OAuth 2.0 is the authorisation framework used by the CSF to control access to API endpoints. It provides token-based security with granular permissions.

How OAuth 2.0 is Used in the CSF

The CSF uses the **Client Credentials flow** — the machine-to-machine (M2M) flow — because CSF communication is between MAP servers, not end users.

Each MAP:

1. Registers with other MAPs during onboarding, receiving a `client_id` and `client_secret`
2. Requests a time-limited access token from the other MAP's OAuth token endpoint
3. Includes the bearer token in API requests to the other MAP's endpoints

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

4. The receiving MAP validates the token's signature, issuer, audience, and expiry on every request

CSF OAuth 2.0 Requirements

- All token exchanges MUST use HTTPS/TLS 1.3
- Token scope and lifetime MUST be limited
- Client secrets MUST NOT be exposed in logs or client-side code
- Token signature, issuer, and audience MUST be validated on every request

> OAUTH 2.0
> : mirrors the approach used by TOTSCo for OTS, making it the most compatible option across the industry. MAPs that already have TOTSCo OAuth credentials may be able to reuse them for CSF connectivity.

6.4 API Key Authentication

API key authentication is a simpler mechanism where a client includes a unique secret key in each request. In the CSF context:

- API keys are recommended for **MAP-to-CP** communication, where full OAuth 2.0 would be excessive
- API keys MUST NOT be used for **MAP-to-MAP** communication over the CSF — OAuth 2.0 is mandatory
- All API key exchanges MUST use TLS 1.3
- Keys MUST be stored securely, rotated regularly, and granted minimal access

6.5 PKI and DKIM Message Signing

Public Key Infrastructure (PKI) provides the cryptographic framework for message signing and verification. The CSF implements PKI through DKIM (DomainKeys Identified Mail, RFC 6376), a standard originally designed for email authentication but ideally suited to federated, distributed messaging.

Why DKIM?

DKIM is ideal for the CSF because it:

- Provides **message integrity** — the message body cannot be tampered with undetected
- Provides **domain-level authentication** — the signature confirms which domain (and therefore which CP) authorised the message
- Uses **DNS for public key distribution** — no centralised certificate authority is needed

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- Gives **CPs control** — a CP's DNS records determine which MAP is authorised to sign on their behalf
- Is a **proven standard** used at internet scale for email authentication

How It Works (Conceptual)

1. When a MAP sends a message, it creates a cryptographic hash of the message body and selected headers
2. The hash is encrypted (signed) with the source CP's **private key** — held securely by the MAP
3. The signature is placed in the HTTP header `X-CSF-SIGNATURE` — the message body is not modified
4. The receiving MAP retrieves the source CP's **public key** from a DNS TXT record
5. The receiving MAP recalculates the hash and verifies it against the decrypted signature
6. If the hashes match, the message is authentic and unaltered; if not, the message is rejected

DNS Records for Each CP

Each CP maintains two DNS TXT records that establish their identity and MAP association:

- **`[RCPID]._domainkey.[cp-domain]`** — Contains the CP's public key for signature verification
- **`[RCPID]._mapkey.[cp-domain]`** — Contains the URL of the CP's MAP message endpoint

These records are controlled by the CP, ensuring that:

- Only the MAP indicated in DNS can sign messages on the CP's behalf
- The CP can change MAP by updating these DNS records
- Any receiving MAP can independently verify the message origin without relying on a central authority

> CSF BEST PRACTICE — DNS INDEPENDENCE.
> A CP SHOULD NOT delegate management of the DNS zone containing `_domainkey` / `_mapkey` records to its MAP. The whole point of the DNS-anchored model is that the CP retains unilateral control of *who is authorised to sign for it* and *which MAP routes its traffic*. If a CP delegates the DNS zone to its MAP, moving to a different MAP — or becoming a MAP of 1 — requires cooperation from the MAP the CP wishes to leave, which can be slow, contested, or commercially unattractive for the outgoing MAP. CPs SHOULD therefore host their own DNS or use an independent DNS provider with whom they have a direct commercial relationship that is not entangled with their MAP

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

> relationship. This keeps the CP's right to change MAP exercisable at any time, by the CP alone.

> CSF BEST PRACTICE — LOCAL DNS SERVER.
> To ensure the DNS records used to verify CSF messages reflect the live public records as closely as possible, MAPs **SHOULD** operate a local DNS server that mirrors `_domainkey` and `_mapkey` records in near-real-time, observing the public TTLs published by each CP. Core standard: **[RFC 6303 — Locally Served DNS Zones](https://datatracker.ietf.org/doc/html/rfc6303)**. This is a common operational pattern (recursive resolver pinned to authoritative servers; or stub bypassing OS resolver cache) and imposes no new infrastructure cost. Detailed rationale and impacts-if-not-implemented are in Part 2: PKI & DKIM Signing §3.3.

> FOR IMPLEMENTERS
> : Detailed DKIM implementation guidance, worked code examples (Java and C#), CSF HTTP headers, signing algorithms, and error codes are provided in Part 2: PKI & DKIM Signing.

6.6 Security Summary

The CSF protects every interaction with:

- **TLS 1.3** — encrypting all data in transit between MAPs
- **OAuth 2.0** — controlling who can access which endpoints
- **PKI/DKIM** — verifying that every message is authentic, unaltered, and authorised by the source CP

No single layer is sufficient on its own. Together, they provide **confidentiality** (data cannot be read), **authentication** (identities are verified), **authorisation** (access is controlled), and **integrity** (content cannot be tampered with).

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

7. Onboarding

7.1 Overview

The CSF uses a peer-sponsored onboarding model. There is no centralised registration authority. Instead, existing MAPs take responsibility for validating and onboarding new participants, ensuring quality and compliance without the cost and complexity of a central governing body.

A minimum of **two MAPs** is sufficient for a basic bilateral exchange. This eliminates the need for a "big bang" approach — MAPs and CPs can be onboarded incrementally.

7.2 Onboarding a New MAP

7.2.1 Sponsorship Framework

New MAPs are onboarded through a structured, multi-phase process guided by a **Sponsor MAP** — an existing, onboarded MAP identified via FIFO rotation. The candidate initiates contact: it discovers the **current designated Sponsor** either from the TAG-published sponsor-registration page (URL to be published by the TAG) or by reading any existing MAP's CP Registry and locating the entry with the most recent `nextSponsorUpdated` timestamp (see the standalone *MAP Onboarding & Testing Process* §3.4 for the detailed mechanism).

Key principles:

- **Next Sponsor MAP**: Uses `nextSponsor` — a forward pointer that says "when I accepted my engagement, I designated X as next." The current sponsor is simply the MAP with the most recent `nextSponsorUpdated` timestamp. No consensus, no propagation of a shared value — just read the most recent write.
- **Capped costs (sponsor fees are free by default)**: Sponsor MAPs are **not expected to charge** for onboarding another MAP. There is an option to charge a fee when significant time and effort is required, capped at **£3,000 + VAT** for a `traditional` (multi-CP) engagement and at **£350 + VAT** for a `single-cp` (MAP-of-1) engagement. The option exists because the CSF expects the Sponsor MAP to apply a **quality gate** — all checks and baseline testing completed to TAG standards. The actual fee is a negotiation between the Sponsor MAP and the new MAP based on how much support is required. A MAP is only charged **once** for onboarding, regardless of how many industry protocols it eventually carries. A set of **shared test endpoints** will be created over time to further reduce sponsoring effort and cost. See §7.4.5 for the `single-cp` fast-track.
- **Gatekeeper role**: The Sponsor MAP acts as a gatekeeper to the CSF network, ensuring operational performance and compliance.
- **Evolving criteria**: The TAG continuously updates entry criteria to reflect the latest maturity levels and address known issues identified through industry testing and operational working groups.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

7.2.2 On-Ramp Process

The onboarding process follows five phases:

> 1. SELECT SPONSOR MAP
> **→ 2. Initial Assessment → 3. Integration & Testing → 4. Operational Validation → 5. Full Operations**

Phase 1 — Sponsor MAP Selection and Agreement

- The new MAP identifies and selects an existing MAP as its sponsor (or is assigned one via rotation)
- The Sponsor MAP verifies the request is legitimate
- Both parties agree on technical, operational, and compliance criteria

Phase 2 — Initial Assessment and Registration

- The Sponsor MAP conducts a capability assessment covering technical readiness, security, and compliance
- The new MAP registers essential details: endpoint URLs, PKI credentials, and directory synchronisation preferences

Phase 3 — Integration and Testing

- Secure communication channels are established using OAuth 2.0 and PKI
- Credentials at this stage provide **provisional/restricted access** — directory information should contain limited, obfuscated data to prevent disclosure of client information until testing is complete
- Testing verifies secure, authenticated message exchanges and directory synchronisation
- Once basic connectivity is proven, access levels are raised to **test** — the directory will now contain information to facilitate test communications (with CP consent or using test CPs)

Phase 4 — Operational Validation

- The Sponsor MAP verifies adherence to operational requirements: message delivery reliability, retry mechanisms, and SLA compliance
- Validation tests ensure interoperability with other MAPs and CPs
- Each other MAP performs additional sanity testing as they establish their own OAuth 2.0 connections with the new MAP
- A **72-hour soak** under projected production load is the default exit gate (see the standalone *MAP Onboarding & Testing Process* §6 Phase 4 and §9 Exit Criteria #2). The soak is **relaxed** for `single-cp` (MAP-of-1) engagements — see §7.4.9.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Phase 5 — Full Operations

- The Sponsor MAP formally approves the onboarding and notifies all other MAPs
- Connectivity is upgraded to **full access**
- The new MAP freely shares its complete CP Registry with other MAPs
- The new MAP commences independent, full operational integration

> NOTE
> The detailed testing process, entry criteria, and exit criteria are set out in the separate "New MAP Onboarding & Testing Process" document maintained by the TAG.

7.3 Onboarding a CP

CP onboarding is a commercial matter between the CP and its chosen MAP. The CSF does not prescribe how MAPs select or charge their CPs. However, the CSF requires certain steps to be completed before a CP can exchange messages:

7.3.1 MAP Responsibilities

1. **Allocate a unique RCPID** — A UUIDv4 identifier, verified against the directory to prevent collisions (see RCPID Management)
2. **Generate PKI key pair** — Create the public/private key pair for the CP
3. **Guide DNS setup** — Assist the CP in creating the two required DNS TXT records:

   - `[RCPID]._domainkey.[cp-domain]` — containing the public key
   - `[RCPID]._mapkey.[cp-domain]` — containing the MAP's message endpoint URL

1. **Collect customer data assets** — Ensure the MAP has sufficient data to perform timely Switch Match responses without relying on the CP being online
2. **Validate compliance** — Ensure the CP meets the requirements of the industry process they wish to participate in
3. **Publish in registry** — Add the CP to the MAP's registry with all mandatory fields

7.3.2 When a CP Can Exchange Messages

No messages can be exchanged until other MAPs verify that the CP is correctly associated with the MAP in its directory. This verification happens automatically when MAPs pull the updated registry and cross-reference it against the CP's DNS records.

A MAP can register a new CP and advertise it to other MAPs **within seconds**. However, the CP controls when it is ready to exchange by managing its DNS entries and service status (ACTIVE, TEST, SUSPENDED).

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

7.4 MAP-of-1 Onboarding (incl. TOTSCo → CSF Migration)

7.4.1 Definition and scope

A **MAP of 1** is a CP that operates the full MAP stack for its own brand: OAuth 2.0 token endpoint, Letterbox, CP Registry / Directory API, DKIM signing, and the CP's own `_domainkey` / `_mapkey` DNS records. The CP is a CSF peer to every other MAP and pays no transport fees to a third party. This section sets out the onboarding path **specifically for** a CP that wishes to take on the MAP role for itself — and in particular for the common case of a CP currently using **TOTSCo for SforB** (and / or for niche bilateral exchanges, or other non-OTS protocols) who wishes to **move that traffic to the CSF as their transport**.

> OTS IS OUT OF SCOPE OF THIS MIGRATION.
> OTS (One Touch Switch — residential switching) is **mandated by Ofcom to use TOTSCo as the single industry platform**. A CP that operates OTS in the UK MUST continue to use TOTSCo for that traffic. The §7.4 migration applies to **SforB** and other **non-OTS** protocols only — for example niche bilateral exchanges between CPs, or future industry protocols if and when the TAG adds them to the CSF baseline (see §7.4.10). A CP that currently uses TOTSCo for both OTS and SforB can move its **SforB** traffic to a CSF MAP-of-1 while **continuing OTS via TOTSCo unchanged**; the two are operationally independent at both the transport layer and the directory layer.

This section is in addition to §7.2 (sponsor-driven multi-CP MAP onboarding) and §7.3 (CP onboarding by an existing MAP). The technical onboarding mechanism is the **same five-phase on-ramp** as §7.2 — the differences are scope, fee guidance, soak relaxation, and the cutover story.

7.4.2 Strategic context — the CP remains in control

The CSF is the open-standard transport for SforB (and over time other industry protocols). It is the **bit pipe** (see Definitions) for exchanging messages between CPs — a network provider that simply moves data without adding value or examining content. The CP — or MAP of 1 — remains in **direct control of its own customer relationship at all times**: end-consumer data, sales, support, billing, in-flight order state, and all CRM live with the CP, not with the transport. The transport (CSF or TOTSCo) carries the JAM-envelope messages; nothing more.

This is the central reader takeaway: a TOTSCo subscriber moving to the CSF as a MAP of 1 is **changing transport, not changing business**. The CP's brand, customers, **SORs (Switch Order References)** and ServiceIdentities are unaffected by the transport switch. The risk of the move is therefore low — provided the CP is in full operational control of its customer relationship before the cutover begins.

> BRAND UNIQUENESS (INDUSTRY CONSTRAINT).
> The SforB industry process supports a CP brand appearing **once** in the industry directory. This is not a CSF transport rule — it is a property of the industry-process layer (SforB / OTS) above the transport. The practical consequence is that a CP migrating from TOTSCo to a CSF MAP-of-1 **cannot dual-run** the same brand on both directories

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

> simultaneously; the move must be a **planned cutover** (see §7.4.7). The exception is during a brand change — e.g. as part of a merger or rebrand — where the new brand is a new directory entry and the old brand is consolidated under the §8.3 process.

7.4.3 Decision tree — "Should I become a MAP of 1?"

A CP considering self-serve operation on the CSF SHOULD work through the following:

#	FACTOR	"YES" IF…
1	**Technical capability**	The CP can run TLS 1.3, OAuth 2.0 (client credentials), DKIM signing (RSA or Ed25519, using existing libraries — see §7.4.10), JSON registry serving, and CP-controlled DNS for the `_domainkey` and `_mapkey` records.
2	**Infrastructure overlap**	Most of the above (OAuth, DNS, TLS, JSON handling, message-queue and retry) is **already** running in the CP's existing TOTSCo integration — the increment for a MAP-of-1 is small.
3	**DKIM signing readiness**	The CP can integrate a DKIM signing library (RFC 6376 reference implementations exist for Java, .NET, Go, Python, Node, etc. — see §12 Tooling Reference in the standalone handbook). This is a one-off software change, not a recurring cost.
4	**Volume and business case**	The CP's annual TOTSCo subscription + unit charges exceed the CP's projected one-off MAP-of-1 build cost over a payback window the CP regards as acceptable. (See §7.4.10.)
5	**On-call cover**	The CP has, or can establish, the 24×7 incident-response posture required by §9 (Operational Excellence) and the SforB SLAs.
6	**Routing footprint**	The CP knows which other MAPs (or HMAPs like TOTSCo) it needs to exchange with, and is prepared to complete the bilateral tests for those it adds beyond the Sponsor MAP.

If the CP cannot meet factor 1 today but expects to in the medium term, an interim **CP-onboarding-with-a-CSF-MAP** path (§7.3) is the safer choice; the CP can transition to MAP-of-1 later via §8.2 mechanisms. If the CP cannot meet factor 5 either, MAP-of-1 is not appropriate and a managed-MAP relationship should continue.

7.4.4 Pre-migration capability checklist

Before approaching a Sponsor MAP, the candidate CP SHOULD verify:

- [ ] **DNS zone control** — the CP controls the DNS zone(s) in which `[RCPID]._domainkey.[cp-domain]` and `[RCPID]._mapkey.[cp-domain]` will live (or has it under independent commercial control — see §6.5 best practice "keep DNS independent of the MAP").
- [ ] **TLS 1.3 endpoint capability** — the CP has, or can stand up, public HTTPS endpoints (OAuth token URL, Letterbox URL, Directory API URL, Service Status URL).
- [ ] **OAuth 2.0 IdP** — the CP can issue and validate client-credentials bearer tokens. Most CPs already operate one for their TOTSCo integration.
- [ ] **DKIM signing** — the CP can integrate a DKIM signing library and produce a CSF-format `X-CSF-SIGNATURE` header that passes against the public NowYoYo PKI Test Harness.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- [ ] **JSON registry** — the CP can publish a CP Registry that validates against the §5 Directory API schema.
- [ ] **Monitoring and on-call** — the CP has alerting, logging, and an on-call rota suitable for SforB SLAs.
- [ ] **CRM authority** — the CP's own CRM is the authoritative source of customer state (so the transport change does not depend on the outgoing hub).
- [ ] **In-flight order list** — the CP can enumerate the in-flight orders open on its TOTSCo subscription as of the planned cutover date (used to plan §7.4.7).

7.4.5 Sponsor MAP programme — fast track for single-CP

Sponsor MAPs are **encouraged** to use the `single-cp` engagement type (tracked as `type: "single-cp"` in `sponsorship.active[]`, see the standalone handbook §3.4) as a positive industry-development offering. The CSF treats existing-MAP sponsorship of TOTSCo-subscriber CPs as a **first-class adoption mechanism**: it grows the open network, gives the sponsor commercial visibility with a new peer, and gives the candidate a low-friction route in.

Differences for a `single-cp` engagement:

ASPECT	TRADITIONAL SPONSORSHIP	`SINGLE-CP` SPONSORSHIP
Fee	Free by default; capped at **£3,000 + VAT** when significant support is required	Free by default; capped at **£350 + VAT** when significant support is required (the absolute ceiling — the reduced scope, single CP, no multi-tenant DKIM, smaller registry, typically keeps the actual negotiated fee well below the cap)
Phase 4 soak	72 hours at projected production load	**Soak waived** for the single-CP MAP-of-1 case — see §7.4.9
Bilateral ≥50% coverage in Phase 4	Required	**Waived** — only the bilateral tests the candidate intends to route to are required (see §7.4.9)
Speed	Weeks to months	Days to weeks once exit gate is met

This is not a quality compromise: the candidate must still pass §8.1 – §8.9 of the standalone handbook against the Sponsor MAP and the NowYoYo PKI harness. What's relaxed is the **production-scale validation step**, on the basis that a MAP-of-1 is **a CP managing its own transport — the operational risk is internalised by the CP** (per §7.4.2). The CP that runs the MAP-of-1 is, by construction, the same CP who carries the consequence of a defect.

7.4.6 Finding your Sponsor MAP

Two discovery paths are available to a candidate (including before joining the network):

1. **TAG-published registration page** (URL to be published by the TAG) — this is the **primary** entry point. The TAG publishes the current designated Sponsor MAP and their registration contact details on the public TAG website. Updated whenever the rotation advances.
2. **Directory bootstrap** (fallback) — read any existing MAP's CP Registry over HTTPS and locate the `sponsorship` entry with the most recent `nextSponsorUpdated`. This is the in-

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

band mechanism documented in the standalone handbook §3.4 and the only mechanism available before the TAG website is live.

Once the current Sponsor MAP is identified, the candidate contacts the Sponsor MAP's `map/contact[]` registration entry (or the TAG page's registration link). The Sponsor MAP's own onboarding portal handles the rest (Phase 1 — see §7.2.2 and the handbook §6).

7.4.7 The cutover journey — big-bang, CP-controlled

Because the SforB industry directory supports a CP brand only once (§7.4.2 brand uniqueness), the move from TOTSCo to a CSF MAP-of-1 **MUST be a planned cutover** — there is no dual-running window for a single brand. The cutover is **CP-controlled** at every step. Five stages:

1. **Build** (CP timetable; typically weeks). The CP builds and tests its MAP-of-1 stack against the Sponsor MAP's Test Tenant. `map.status=TEST`. The CP's TOTSCo subscription is unaffected. Phases 1 – 3 of §7.2.2 complete.
2. **Exit-gate pass** (CP timetable). Baseline tests §8.1 – §8.9 (handbook) pass against the Sponsor MAP and the NowYoYo PKI harness; Phase 4 relaxations per §7.4.9 satisfied; Sponsor MAP issues Phase 5 approval.
3. **In-flight position** (CP decision). The CP chooses the in-flight policy for its open TOTSCo orders. **Two valid paths** — neither imposed by the framework; the CP picks based on volume and risk:

    - **Drain-first (preferred when feasible)** — the CP stops accepting new switch matches on TOTSCo, lets all open in-flight orders complete or cancel cleanly, then performs the cut with zero in-flight balance. Lower operational complexity at cutover; longer total elapsed time.
    - **CP-managed migration** — the CP cuts when ready and reconciles in-flight orders from its own CRM. SORs and ServiceIdentities **do not need to change** — the CP holds the customer relationship throughout and the transport change is invisible to the underlying business state. If TOTSCo can be persuaded to honour an `rcpidStatusRequest`-equivalent export, happy-path import via the MAP-of-1 applies; otherwise the CRM-reconstruction model from §8.2.4 (CP Emergency Migration unhappy path) applies. The CP carries the operational risk — which is appropriate because the CP is the party in control.

1. **Cutover (big-bang)**. The CP coordinates with TOTSCo to terminate the **TOTSCo SforB** directory entry for the brand as the new CSF MAP-of-1 directory entry goes live. If the same brand also operates OTS via TOTSCo, the **OTS directory entry is unaffected** and continues unchanged — OTS is Ofcom-mandated to TOTSCo (see §7.4.1) and is out of scope of this migration. **TTL pre-staging** is used on the CP's `_mapkey` and `_domainkey` to keep the observed outage to single-digit minutes (the standard enterprise DNS-change routine: lower the TTL ≥24h before the cut-over, swap, then restore — see Part 2 §3 *PKI & DKIM Signing* for the canonical clause). The new CSF UUIDv4 RCPID is

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

allocated per §7.4.8 — the legacy TOTSCo **SforB** `Rxxx` is retired (not migrated); any OTS `Rxxx` for the same brand on TOTSCo is independent and untouched.

2. **TOTSCo wind-down (SforB only)**. The CP closes its **TOTSCo SforB** participation per the TOTSCo contract terms (notice period, final invoice). If the CP continues to use TOTSCo for OTS, the **OTS subscription / participation continues unaffected** — only the SforB element is wound down. Audit retention rules per §9 apply.

> THE RISK IS LOW PRECISELY BECAUSE THE CP IS IN CONTROL.
> The CSF (and TOTSCo before it) is the transport, not the business. End-consumer relationships, SORs, ServiceIdentities, billing, support — all of these live in the CP's CRM and are unaffected by the cut.

7.4.8 RCPID allocation rule

A CSF MAP-of-1 is allocated a **new UUIDv4 RCPID** in the standard way (see §8 RCPID Management) — the legacy TOTSCo `Rxxx` 4-character RCPID is **not reused, not migrated, and not preserved as the CSF RCPID**. The two identifiers are independent.

For OTS-side historic reporting (post-cutover queries that reference the legacy `Rxxx`), the CP retains its TOTSCo records under TOTSCo's own retention regime; the CSF retention applies only to messages exchanged over the CSF.

7.4.9 Phase 4 relaxation for MAP-of-1

For `single-cp` (MAP-of-1) engagements, Phase 4 (Operational Validation) is modified as follows:

§7.2.2 PHASE 4 ELEMENT	TRADITIONAL MAP	MAP-OF-1 (`SINGLE-CP`)
72-hour soak under projected production load	**Required** (handbook §6 Phase 4 / §9 #2)	**Waived** — a single-CP MAP has no meaningful production-scale traffic to soak, and the CP carries the operational risk directly
Bilateral connectivity with ≥50% of onboarded MAPs	**Required** (handbook §9 #4)	**Waived** for the ≥50% target. **Required** for any MAP / HMAP the candidate intends to exchange with at go-live, including TOTSCo as HMAP
Retry & delivery policy tests (§8.7)	Required	**Required — unchanged**
Conflict resolution & DNS tests (§8.8)	Required	**Required — unchanged**
Security & negative edge cases (§8.9)	Required	**Required — unchanged**
§8.10 V-01 (72-hour soak)	Required	**Waived** with V-02 – V-05 still required

The Sponsor MAP records the waivers in the Phase 5 approval record. The TAG MAY require additional evidence if subsequent operational incidents suggest a single-CP MAP-of-1 has been under-tested in this respect.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

7.4.10 Bilateral testing of additional protocols

The CSF transport is process-agnostic. A MAP-of-1 — like any other MAP — MAY use the CSF as the transport for **niche routing** or future industry protocols beyond SforB.

- **SforB** — if the MAP-of-1 supports SforB, the §8.6 SforB baseline tests are **mandatory** for Phase 3 sign-off (no relaxation for `single-cp`).
- **Niche routing / bilateral protocols** — for any protocol the candidate adds beyond what's currently in the baseline, the candidate and its bilateral counterparty MUST **buddy-test** the protocol against each other once the Sponsor MAP has signed off the CSF transport baseline. This applies equally to traditional and `single-cp` engagements.
- **Future industry protocols** — when a future protocol (for example, **OTS Porting**, or a future bilateral notification flow) is brought onto the CSF, the TAG will add a §8.x baseline test set for it (see the standalone handbook §8 "Scope and Extensibility"). Once published, a MAP that adopts that protocol — including a MAP-of-1 — MUST pass the corresponding baseline before Sponsor sign-off.

This means a MAP-of-1 supporting only SforB completes the §8 baseline once; a MAP-of-1 that subsequently adds a niche bilateral protocol completes that protocol's baseline (if published) and its bilateral tests at the point it adds the support.

7.4.11 Cost model — qualitative

The MAP-of-1 cost story (qualitative; specific numbers vary by candidate):

- **Most of the operating infrastructure is already running.** A CP currently subscribing to TOTSCo already operates OAuth 2.0 client integration, DNS records, TLS endpoints, retry queues, and JSON message handling. The MAP-of-1 stack reuses these.
- **DKIM signing is the principal new component.** DKIM is RFC 6376 — there are mature reference implementations in every major language (Apache James jDKIM for Java, MailKit for .NET, etc. — see standalone handbook §12 Tooling Reference). The cost is a **one-off software-integration exercise**, not a recurring infrastructure spend.
- **Directory API serving is small.** A single-CP registry is a few-kilobyte JSON document. Any web framework can serve it.
- **Sponsor fee is free by default; one-off and capped** at **£3,000 + VAT** for traditional engagements, or **£350 + VAT** for `single-cp` engagements, only when significant Sponsor MAP support is required (see §7.4.5). Charged once per MAP regardless of how many protocols it eventually carries.
- **The savings are the recurring TOTSCo charges.** The MAP-of-1's running cost is principally a small increment over what the CP was already paying for its TOTSCo integration; the operating saving is the **removal of TOTSCo subscription and per-unit fees**. A CP whose annual TOTSCo spend exceeds the one-off MAP-of-1 build plus the annual marginal cost of the additional CSF-specific components has a positive business case.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

CPs SHOULD produce their own business case using their own commercial inputs. The framework provides no transport fees of its own.

7.4.12 What a MAP-of-1 *does not* lose

A CP moving from TOTSCo to a CSF MAP-of-1 retains:

- Its end-consumer relationship and CRM
- Its SORs and ServiceIdentities (the transport change does not propagate down)
- Its existing OAuth 2.0 / DNS / TLS infrastructure (most of it is reused)
- Its industry contact-details discoverability — the CSF Directory carries CP-to-CP, CP-to-MAP, MAP-to-MAP, and MAP-to-CP contact entries (phone, email, ticketing URL, purpose tags, availability hours). **The TOTSCo OTS CP-to-CP tool is not mandated for SforB**; the CSF registry contact entries replace the need for a specialised industry tool.
- Its right to come back — the CP can move from MAP-of-1 to a managed-MAP relationship later via the §7 normal transition process or, if necessary, §8.2 CP Emergency Migration.

7.5 Testing in Production

The CSF supports testing in production during onboarding:

- All test traffic MUST be explicitly marked as test in the audit record held in the JAM envelope
- The onboarding MAP MUST override the test flag until the CP has passed exit criteria — even if the CP sets it
- A published set of test data assets is available for entry criteria testing, performed locally with the MAP using a Virtual Test Host
- A hidden set of unit tests forms the exit criteria before the CP is granted production status
- PKI verification tools are available to assist with setup and testing

> NOTE
> : Not all CPs will have SIT/UAT environments. The CSF is designed to support continuous onboarding throughout core business hours, with strict protocols and test-flagging to prevent test traffic from affecting production operations.

7.6 Memorandum of Understanding (MoU)

MAPs and CPs that collaborate on the CSF are bound by a Memorandum of Understanding that outlines the terms of participation. The MoU is designed to be simple to join, reflecting the CSF's nature as an open standard that is licence-free for any CP or MAP to use for any purpose.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

8. RCPID Management

8.1 What is an RCPID?

A Retail Communications Provider Identifier (RCPID) is a unique identifier assigned to a CP brand for the purpose of message exchange. Each RCPID maps 1:1 to a human-readable brand name that appears in search and drop-down lists used by CPs when initiating a Switch Match.

8.2 Why UUIDv4?

The CSF adopts **UUIDv4** (RFC 4122) as the RCPID format, replacing the 4-character alphanumeric codes used by OTS.

Problems with the OTS Format

The OTS format (`Rxxx` — e.g., RGXD, RYHC) has several limitations:

- **Collision risk**: With only 4 characters, the probability of duplication increases as more CPs join for business switching (expected tenfold increase over OTS)
- **Human confusion**: Similar codes are easily mistaken (e.g., RYHC and RYCH are different brands but commonly confused — and they can exist in the same MAP)
- **MAP coupling**: If prefixes are tied to MAPs, changing MAP requires changing RCPID, which is costly and disruptive
- **Scalability**: The format does not scale for business switching where CPs may have multiple brands requiring separate identifiers

Benefits of UUIDv4

UUIDv4 addresses all of these issues:

- **Virtually zero collision risk**: With 122 bits of randomness, the odds of collision between any two UUIDs are 1 in 2.71 x 10^18. One would need to generate 1 billion UUIDs per second for 85 years to have a 50% chance of a single collision.
- **Self-allocation**: MAPs can generate RCPIDs independently without central coordination
- **No human bias**: Machine-generated identifiers eliminate human-readable confusion
- **Portability**: The RCPID stays with the CP for life, regardless of which MAP they use
- **Data continuity**: All historical data (time-series, audit records, reporting) remains consistent as CPs move between MAPs

Example RCPID: `d8322d80-92c8-4906-a408-f1d2daf7e03d`

Compatibility with OTS

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

The UUIDv4 format does not preclude any CP or MAP (including TOTSCo) from continuing to use the OTS `Rxxx` format internally. MAPs can maintain a mapping between their internal codes and the UUIDv4 RCPID used for SforB message exchange.

The SforB Message Specification defines the source and destination identity fields as strings with a maximum length of 256 characters. UUIDv4 (36 characters including hyphens) fits well within this constraint.

8.3 Allocation Process

1. The MAP generates a UUIDv4 for the new CP brand
2. The MAP provisionally allocates it to the CP
3. Before going live, the MAP pulls all other MAP directories and checks for collisions
4. If no collision is detected, the MAP publishes the RCPID in its registry
5. If a collision is detected (extremely unlikely), the MAP generates a new UUID and repeats

> RESPONSIBILITY
> : It is the MAP's responsibility to ensure no duplication or collisions occur. The MAP MUST check the generated ID against the directory before allocating it to a CP and advertising it as live.

8.4 RCPID Lifecycle

Allocation

An RCPID is allocated to a CP brand and remains associated with that brand **for life**. This helps with:

- CPs moving between MAPs
- Future brand consolidation or growth
- Mergers and acquisitions
- Consistent time-series data and historical reporting across all MAPs

MAP Transitions

When a CP moves between MAPs:

1. The RCPID **does not change** — it remains constant and unique across the network
2. The new MAP adds the existing RCPID to its CP Registry
3. The old MAP removes the RCPID from its CP Registry
4. Other MAPs detect the change through CP Registry synchronisation and DNS verification

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

This ensures seamless continuity for in-flight orders and network-wide consistency.

Retirement

When a CP brand is dissolved or merged, the RCPID is retired and retained in an archived state for historical reporting purposes. It MUST NOT be reused for a different brand. While there is no centralised record each MAP will be responsible for hold a legagcy record and avoiding the risk of re-using for a nominal period of time. Once all inflight orders are resolved and SORs expired then there is a low risk asociated with reuse. It is also very unlikely, given UUID v4 size and the method suggested for picking new ones, that anyone will pick a previously used RCPID.

8.5 Handling Exports During Transitions

When a CP moves to a new MAP, the in-flight order data must be transferred:

1. **Export with ID mapping**: On a valid "RCPID Status" request, the old MAP provides a standardised JSON export of all in-flight orders for the specified RCPID
2. **Validation**: The request is validated by checking that the RCPID's DNS entry now points to the MAP making the request
3. **CP Registry updates**: The old MAP removes the CP from its CP Registry before responding, ensuring no duplicate entries exist
4. **Verification**: The new MAP verifies the accuracy of the RCPIDs and ensures no collisions

> NOTE
> For the complete transition process including in-flight order handling, see Part 2: CP Transitions.

8.6 Summary

ASPECT	OTS (LEGACY)	CSF
Format	4-character alpha (Rxxx)	UUIDv4 (36 characters)
Allocation	Centralised or MAP-specific	MAP self-allocates
Collision risk	Moderate (grows with scale)	Negligible (1 in 2.71 x 10^18)
Portability	Tied to MAP (prefix-based)	Follows the CP for life
Human readability	Confusing (RYHC vs RYCH)	Not human-readable (by design)
Scalability	Limited for business switching	Unlimited

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

9. Operational Excellence

9.1 Core Principles

The CSF is built on three operational cornerstones:

- **Security**: Strong encryption, robust authentication, and digital signing
- **Efficiency**: Accurate and timely telecom service transitions
- **Reliability**: Resilient architecture ensuring continuous availability

9.2 Availability Target

MAPs MUST target **99.95% availability** (three nines & a five) for switching services. This equates to 21 minutes and 55 seconds of downtime per month, or about 4 hours and 23 minutes of downtime per year and aligns with TOTSCo Availability Target.

9.3 High Availability and Redundancy

MAPs SHOULD:

- Deploy **multiple geographically distributed** instances (subject to GDPR) to eliminate single points of failure
- Implement **automatic failover** mechanisms capable of instantly transitioning operations to backup systems without noticeable service disruption
- Maintain **fully redundant network paths** and infrastructure components

9.4 Continuous Operations During Maintenance

MAPs MUST implement rolling updates and upgrades to ensure that maintenance activities — including software updates and security patching — do not impact service continuity.

Recommended deployment strategies:

- **Blue-Green Deployment**: Maintain two identical production environments. Deploy updates to the inactive environment, validate, then seamlessly switch traffic. This enables rapid rollback if issues are discovered.
- **Canary Deployment**: Gradually roll out updates to a small subset of traffic, monitor performance, and incrementally increase exposure based on positive outcomes.

MAPs SHOULD clearly communicate planned maintenance activities — even when designed to have no service impact — to maintain transparency and stakeholder confidence.

9.5 Monitoring and Alerting

MAPs MUST provide:

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- **Real-time, proactive monitoring** with automated alerting capable of detecting and addressing issues before they affect availability
- **Mandatory status indication** in the MAP Registry when an incident occurs, including clearly recorded planned changes highlighting any periods of potential risk

MAPs SHOULD provide:

- A transparent, real-time **status dashboard** accessible by connected providers, showing platform health and operational metrics

9.6 Disaster Recovery

MAPs MUST:

- Regularly **test and validate** disaster recovery procedures, including full failover scenarios and restoration processes
- Establish **Recovery Time Objectives (RTO)** and **Recovery Point Objectives (RPO)** that align with the commitment to continuous availability

9.7 Security and Compliance

MAPs MUST:

- Maintain rigorous security standards and apply timely patches without disrupting ongoing service
- Adopt **secure-by-design** principles to ensure resilience includes robust protections against cyber threats
- Conduct **regular security audits** and vulnerability assessments to identify and mitigate risks proactively

9.8 Guaranteed Message Delivery

Pre-Send Verification

Before sending a message, the originating MAP SHOULD verify the resilience status of the receiving MAP using information from the MAP Directory. This includes checking:

- Whether the receiving MAP is currently online
- Whether they are in a known period of planned change or experiencing an incident

Retry Policy

If the originating MAP does not receive a synchronous `HTTP 202` acknowledgement (e.g. it receives a `4xx`, `5xx`, or times out), it MUST:

1. Apply the **1 second connection timeout** and **3 second response timeout** specified in Part 2 §4.7.2.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

2. Automatically retry message delivery using **exponential backoff with jitter** — never a linear or fixed-cadence schedule. The architectural rationale for this is set out in Part 2 §4.7.5.
3. Implement a **circuit-breaker pattern** after N consecutive failures (recommended `N = 5`, tunable per MAP) to stop overwhelming a struggling destination.
4. Honour `Retry-After` on `HTTP 429` / `503` responses.
5. Maintain comprehensive **audit trails** of all message exchanges, delivery attempts, and outcomes.
6. Define a **bounded maximum retry period** — the framework recommends hours, not days. Beyond the window, the sender raises `9008` and the failure escalates to the operational-engagement path in Part 2 §4.7.3: `map/outage[]` publication by the failing MAP, alert on the sender, inter-MAP engagement via `map/contact[]`.

The retry policy applies to **CSF MAP-to-MAP traffic**. Traffic crossing the **HMAP boundary** to TOTSCo follows the GPLB hub schedules on TOTSCo's side — see Part 2 §4.7.4 and the TOTSCo HMAP Integration Guide.

Asynchronous Error Handling

The CSF employs asynchronous error codes (8xxx series) to signal and assist MAPs during message processing. These codes are designed for self-healing and promoting good practices, complementing the industry-standard 9xxx codes used by OTS/TOTSCo.

> **NOTE**
> For details on error codes and message delivery failure formats, see Part 2: Message API.

9.9 Service Level Agreements

Specific SLAs for MAP-to-MAP activity are to be formalised. The following indicative levels have been agreed:

CATEGORY	TARGET
Platform availability	24x7 high availability (queue and retry during unavailability)
Directory freshness	Near-real-time updates (never delete the last compiled directory)
Working hours	Core business hours for support
Planned maintenance	Outside business hours (with objective of minimal planned outages)
MAP-to-MAP response time	1 working day
Priority 1 (MAP unavailable)	Fix within 4 working hours
Priority 2 (CP unavailable)	Fix within 1 working day
Priority 3 (Service degradation)	Fix within 2 working days
Priority 4 (Change requests)	Actioned within 1 working week

9.10 Capacity Planning

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

MAPs should scale their infrastructure to align with their CP base. The following minimum rate calculation has been agreed by the TAG and applies for CSF v2.0:

Messages per second = Total Switchable Asset Base / 1,000

With an **absolute minimum of 2 messages/second**.

What counts as a Switchable Asset

The **Switchable Asset Base** is the total count of customer services across all of a MAP's CPs that can be switched as part of the **Switching for Business (SforB)** protocol. Two service types are in scope:

- **IAS — Internet Access Service.** Broadband or internet connectivity provided to end customers — e.g. fibre-to-the-premise (FTTP), fibre-to-the-cabinet (FTTC), cable broadband, fixed-wireless access. Defined in the Switching General Conditions and reflected in the CSF Definitions and Terminology.
- **NBICS — Number-Based Interpersonal Communications Service.** Voice telephony provided to end customers via an assigned number — e.g. traditional landline, VoIP, hosted business voice. Also covered by the Switching General Conditions.

A "switchable asset" is therefore one IAS line *or* one NBICS line. A single business customer can therefore contribute multiple assets to the count (one IAS + one NBICS = two switchable assets for that customer). The Switchable Asset Base aggregates this across every CP a MAP serves.

This figure is the input to the capacity formula above and to several SLA / governance calculations elsewhere in the CSF.

Examples:

- MAP with 5,000 switchable assets: 5,000 / 1,000 = 5 msg/s
- MAP with 20,000 switchable assets: 20,000 / 1,000 = 20 msg/s

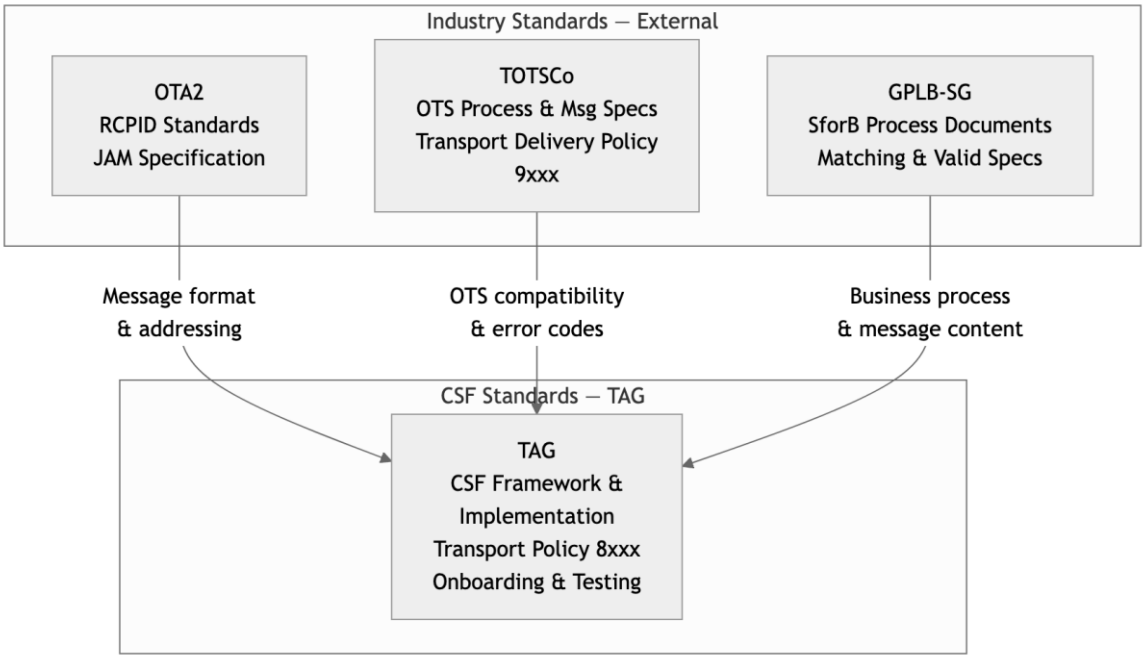
MAPs can enhance acceptance rates using hardware acceleration and internal queuing, as SLAs for business switching are significantly longer than the 60-second Switch Match response times required by OTS.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

10. Governance

10.1 Standards Landscape

The CSF operates within a multi-layered governance structure. Each layer has its own standards body, documentation, and change control process. The CSF leverages these existing standards rather than duplicating them.



10.2 Standards Bodies

10.2.1 OTA2 — Office of the Telecommunications Adjudicator

Accountable for:

- RCPID Standards
- JSON Asynchronous Messaging Specification (JAM Spec)

The industry has established RCPIDs and the JAM envelope as standards that MUST be implemented by all current and future transport providers, including TOTSCo and the CSF. These standards enable consistent addressing and routing of messages. The use of specific envelope fields may vary across industry processes (OTS, SforB) but the fields themselves are standardised.

All documentation is published on the OTA2 website.

10.2.2 TOTSCo — Telecoms One Touch Switching Company

Accountable for:

- All OTS-related Message Process and Message Specification documentation

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- OTS Best Practice Guidance
- Transport Delivery Policy and Response Codes (9xxx)
- TOTSCo Hub API Specification v2.0

The Industry Process Group (IPG) and Operations Group (OG) generate original documentation and proposed changes. All documentation is under change control and published on the TOTSCo website.

10.2.3 GPLB-SG — GPLB Steering Group

The **GPLB Steering Group (GPLB-SG)** is an independent body of industry stakeholders working alongside the OTA2. It guides and facilitates the steering meetings and the publication of the Switching for Business (SforB) process documentation via FCS. The group retains the historical "GPLB" label even though the underlying **process** has been renamed from Gaining Provider Led Business switching to **Switching for Business (SforB)** to better describe the protocol.

Accountable for (via FCS publication):

- Switching for Business (SforB) Process documentation
- Switching Open Orders specification
- Matching Summary and Validation Process
- **SforB Response Codes** and Asset List Guidance
- Customer Matching Guidance
- Best Practice Guide for Avoidance of Erroneous Transfers
- SLAs

All documentation is under change control from 30 September 2025 and published on the FCS website.

10.2.4 TAG — Telecom Technical Architecture Group

Accountable for:

- Connected Services Framework (CSF) — this documentation
- CSF Transport Delivery Policy and Response Codes (industry 9xxx + CSF 8xxx enhancements)
- Testing and Onboarding of MAP documentation
- Hub Implementation Overview Plan

The TAG maintains autonomy to manage and iterate specific areas of the CSF. All changes to CSF documentation are undertaken by the TAG group, which meets via its steering group call **weekly**.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

10.3 Change Control

CSF Documentation

All CSF documentation builds on existing OTS message exchange standards, including directory elements and TOTSCo 9xxx codes, to maintain consistency and avoid misunderstandings. Where additional response codes are required for security and signalling between MAPs and CPs, these are introduced using the **8xxx series**.

Changes to the CSF follow this process:

1. **Proposal**: Any TAG member can propose a change
2. **Review**: The TAG steering group reviews the proposal at its weekly meeting
3. **Consensus**: Changes require consensus among TAG members
4. **Publication**: Approved changes are published and versioned
5. **Adoption**: MAPs adopt changes incrementally through API versioning and feature releases

Industry Process Documentation

Changes to industry process documents (SforB process, JAM Spec, RCPID standards) follow the change control processes of their respective standards bodies (GPLB-SG, OTA2, TOTSCo). The CSF does not control these processes but references them.

10.4 Memorandum of Understanding

The MAPs and CPs that participate in the CSF are bound by a Memorandum of Understanding (MoU). The MoU:

- Makes it simple to join and contribute to the CSF
- Reflects the CSF's nature as an open standard
- Is licence-free for any CP or MAP to use for any purpose
- Outlines the collaborative responsibilities of participants

10.5 Decentralised Governance

The CSF deliberately decentralises governance responsibilities:

- **No centralised administrator**: Controls are distributed to CPs and MAPs
- **Self-regulation**: The framework promotes good behaviour and discourages bad behaviour through its design (e.g., DNS-based CP control, PKI verification, directory transparency)
- **Commercial independence**: MAPs and CPs are free to establish their own commercial relationships; the CSF has no impact on these arrangements
- **Cost reduction**: Decentralising basic functions reduces costs for all parties and makes message exchange more transparent

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

10.6 Governance Controls

The governance controls below were ratified by the TAG steering group on **2026-04-29** and form the operational baseline for the CSF's decentralised model. Each control leverages the framework's existing peer-to-peer architecture and the TAG's weekly steering group, without introducing heavy central administration.

10.6.1 TAG Decision-Making and Voting

The TAG steering group operates by consensus wherever possible. Where consensus cannot be reached, the following formal procedure applies:

- **Quorum**: A formal vote MUST be attended (in person or by recorded proxy) by at least **60% of currently onboarded MAPs**. A vote held below quorum is invalid and MUST be rescheduled.
- **Voting weight**: **One MAP, one vote**, regardless of MAP size, footprint, or seniority. A MAP that operates as a MAP-of-1 carries the same single vote as a MAP serving many CPs.
- **Voting threshold**: A simple majority of votes cast is sufficient for operational decisions. Specification changes (additions to or breaking changes in the CSF documents) require a supermajority of votes cast.
- **Decision records**: All TAG decisions MUST be recorded with the date, attendees, proposal summary, vote tally, outcome, and dissenting views. These records are accessible to all MoU signatories.
- **Observer status**: Organisations that are not yet MAPs but have a legitimate interest (e.g., prospective MAPs, industry bodies, OTA2, TOTSCo) MAY attend TAG meetings as observers. Observers contribute to discussion but do not count toward quorum and have no voting rights.

10.6.2 MAP Compliance and Periodic Review

Onboarding validates a MAP at a point in time, but ongoing compliance must also be assured:

- **Annual self-certification**: Each MAP SHOULD submit an annual self-certification to the TAG confirming continued compliance with the CSF requirements, including PKI key hygiene, SLA adherence, and CP Registry accuracy.
- **Peer review**: The TAG SHOULD establish a lightweight peer review process where MAPs periodically verify each other's CP Registry accuracy, DNS record consistency, and endpoint availability. This can be automated through existing registry collection and DNS verification mechanisms.
- **Evolving exit criteria**: As the TAG updates onboarding entry and exit criteria to reflect new use cases and lessons learned, existing MAPs SHOULD be required to demonstrate compliance with material changes within an agreed adoption window.

10.6.3 Version Adoption Policy

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

API versioning enables incremental change, but without adoption timelines the network could fragment:

- **Minimum support window**: When a new CSF version is released, MAPs MUST continue to support the previous version for **six months** from the release date, to allow all participants to upgrade.
- **Deprecation notice**: The TAG MUST provide at least 3 months' notice before deprecating a CSF version.
- **Adoption tracking**: The TAG SHOULD track which CSF version each MAP supports (visible in the CP Registry) and follow up with MAPs that have not adopted within the agreed window.

10.6.4 Dispute Resolution

Disputes between MAPs (e.g., over CP ownership, message delivery failures, or onboarding decisions) need a resolution path that does not require legal action as a first resort:

- **Tier 1 — Direct resolution**: The MAPs involved attempt to resolve the dispute directly using the contact details published in their CP Registries.
- **Tier 2 — TAG mediation**: If direct resolution fails within 5 working days, either party can escalate to the TAG steering group, which will mediate at its next meeting. The TAG may appoint a subset of uninvolved MAPs to review the facts.
- **Tier 3 — Independent adjudication**: If TAG mediation fails, the dispute MUST be referred to an **external, independent body** appointed by the TAG to act as the standing Tier-3 adjudicator for the CSF. The adjudicator's decision is binding on the parties to the dispute. The TAG's **preferred candidate** for this role is the **Office of the Telecommunications Adjudicator (OTA2)** — see §10.2.1 — given its established independence and history of binding industry dispute resolution; the appointment is **subject to formal acceptance** by the candidate body. The TAG will engage the appointed adjudicator directly and provide the case file, supporting evidence, and the result of Tier 1 and Tier 2 attempts. Until an external Tier-3 adjudicator has formally accepted the role, the TAG SHALL document the route used for any Tier-3 case (typically bilaterally-agreed commercial arbitration between the parties) so the audit trail is unaffected.
- **Interim measures**: During any dispute, both MAPs MUST continue to exchange messages normally. A dispute MUST NOT be used as a reason to block or degrade service to CPs.

10.6.5 Incident Response Coordination

While each MAP manages its own security, coordinated response to network-wide incidents is essential:

- **Security contact**: Each MAP MUST publish a dedicated security contact in its CP Registry (separate from general support) for reporting vulnerabilities and incidents.
- **Coordinated disclosure**: If a MAP discovers a vulnerability that affects the CSF specification or other MAPs, it MUST notify the TAG within 24 hours. The TAG will coordinate disclosure and remediation across all MAPs.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- **Breach notification**: If a MAP suffers a data breach that could affect CP data or message integrity, it MUST notify all connected MAPs, the TAG, and affected CPs within **72 hours of becoming aware**, in alignment with the wider industry standard set by UK GDPR / Data Protection Act 2018 Article 33 personal-data breach notification. Notifications MUST include the nature of the breach, categories and approximate volume of data affected, the likely consequences, and the mitigations taken or proposed.
- **Post-incident review**: Significant incidents SHOULD trigger a post-incident review at the TAG steering group to identify systemic improvements.

10.6.6 MAP Suspension and Removal

The CSF must have a clear, proportionate process for dealing with MAPs that persistently fail to meet their obligations or act maliciously:

- **Warning**: The TAG issues a formal written warning identifying the non-compliance and a remediation deadline (typically 10 working days).
- **Probation**: If the MAP fails to remediate, the TAG may place the MAP on probation. During probation, the MAP's status is flagged in other MAPs' Master Registries, and the MAP loses its voting rights in TAG decisions.
- **Suspension**: For serious or persistent non-compliance, the TAG may vote to suspend a MAP. Suspended MAPs' CPs are treated as if the MAP were in distress — the CP emergency migration process (see Commercial Scenarios) applies, and CPs are encouraged to move to another MAP.
- **Removal**: A suspended MAP that does not remediate within 30 days may be permanently removed from the CSF by TAG supermajority vote. The MAP's CP Registries are no longer collected by other MAPs.
- **CP protection**: At every stage, the priority is CP continuity. No enforcement action against a MAP should leave CPs unable to exchange messages for longer than necessary. The CSF's built-in ability to migrate CPs — including their in-flight orders — between MAPs via the RCPID Status export process (see CP Transitions) is an integral safeguard here. Where a MAP is suspended or removed, the TAG SHOULD proactively coordinate with affected CPs and receiving MAPs to ensure that the CP Emergency Migration Process (export inflight orders) is initiated promptly, in-flight order exports are completed before the MAP is taken offline, and CPs are re-established on new MAPs with minimal disruption to end consumers.

10.6.7 Anti-Competitive Safeguards

The CSF's open, free-to-use nature must be actively protected:

- **No exclusionary behaviour**: No MAP or group of MAPs may act to exclude a legitimate new entrant from the CSF. Sponsor MAPs MUST NOT unreasonably refuse or delay onboarding.
- **No preferential routing**: MAPs MUST NOT give preferential treatment to their own CPs' messages over messages routed on behalf of other MAPs' CPs.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- **No information misuse**: Information obtained from other MAPs' CP Registries (e.g., CP lists, contact details, switching volumes) MUST NOT be used for competitive advantage, marketing, or customer solicitation.
- **Telemetry collection**: Every MAP MUST collect telemetry data for all messages sent and received **between `ACTIVE`-status MAPs and on behalf of `ACTIVE`-status CPs only**, including: message type (routingID), date/time, source CP (RCPID), destination CP (RCPID), source MAP, destination MAP, and delivery outcome (success, failure code). Messages where either endpoint MAP carries `map.status = TEST` or `SUSPEND`, where either CP carries `processSupport[].status = TEST` or `SUSPEND`, **or where the JAM envelope's `auditData` carries the explicit test marker (`{ "name": "test", "value": "true" }`)** MUST be excluded from telemetry counted toward industry statistics. This ensures that onboarding traffic, bilateral test runs, and post-incident soak testing never pollute reporting or KPIs. Telemetry data MUST be stored securely and handled in accordance with GDPR and the data retention policies agreed by the TAG. Test-flagged messages MAY still be logged separately for audit and onboarding evidence, but MUST be tagged as such and held outside the production telemetry corpus. See Onboarding & Testing Process §5 for the audit-flag rules and §4 for status semantics.
- **Transparency**: The TAG SHOULD publish an annual summary of CSF network activity — including aggregated message volumes by message type, number of active MAPs and CPs, onboarding activity, and delivery success rates — to demonstrate openness and healthy competition. Each MAP MUST contribute its aggregated telemetry to this summary, **drawn solely from production traffic between `ACTIVE` participants** as defined above. Individual CP-level data, message content, or commercially sensitive patterns MUST NOT be disclosed — only aggregated, anonymised statistics are published. Detaile on this will follow in subsequent revisions for options on how telemetry can be collected and shared without revealing sensitive data or breaching GDPR.

10.6.8 CP Representation

While the CSF primarily governs MAP-to-MAP interactions, CPs are the ultimate beneficiaries and should have a voice:

- **CP feedback channel**: The TAG SHOULD establish a mechanism for CPs to raise concerns or propose improvements, either directly or through their MAPs.
- **CP rights charter**: The CP Rights Charter is published as a standalone document, written in plain language, so CPs can understand their entitlements without reading the full CSF specification. It is derived from the formal requirements in Principles & Requirements — CP Right to Move.

10.6.9 Documentation and Transparency

- **Public documentation**: The CSF specification (this documentation) SHOULD be publicly accessible to any interested party, not restricted to MoU signatories. Transparency builds trust and encourages adoption.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- **Change log**: All changes to the CSF specification MUST be recorded in a versioned change log with dates, descriptions, and the TAG meeting reference where each change was agreed.
- **Archived versions**: Previous versions of the CSF specification MUST be retained and accessible so that MAPs running older versions can reference the specification they implemented against.

10.7 Supporting Documents

The following documents support the CSF and are maintained separately:

DOCUMENT	VERSION	MAINTAINER
MAP Onboarding and Testing	Draft	TAG
PKI Overview	Draft	TAG
Use-Cases and Examples	Draft	TAG
TOTSCo Integration — Implementation Plan	v1.1	TAG
TOTSCo Integration — Overview Slides	v1.0	TAG
JSON Asynchronous Messaging Specification	v1.0	OTA2
Principles of Use for RCPID	v1.0	OTA2
TOTSCo Hub API Specification	v2.0	TOTSCo

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Glossary

This glossary defines the abbreviations and terms used throughout the CSF documentation.

Abbreviations

ABBREVIATION	FULL TERM	DEFINITION
CP	Communications Provider	An organisation participating in an industry process (e.g., OTS, SforB) that creates or consumes messages and sends or receives them via a MAP. CPs include retailers, wholesalers, and agencies providing services on behalf of other CPs.
CSF	Connected Services Framework	A set of open message communication standards that define how MAPs exchange messages securely and efficiently on a peer-to-peer basis, without reliance on a centralised hub.
DDG	Detail Design Group	The original industry group that documented the requirements for the technologies and processes of switching residential customers' broadband and voice services (One Touch Switch).
DKIM	DomainKeys Identified Mail	A message authentication standard (RFC 6376) that uses PKI to verify the sender's domain identity and ensure that the message has not been altered in transit. The CSF adapts DKIM for HTTP message signing between MAPs.
DNS	Domain Name System	The internet system that resolves domain names to IP addresses. In the CSF, DNS TXT records are used to publish CP public keys and MAP associations.
GPLB	Gaining Provider Led Business Switching	The original name for the industry **process** for switching business telecommunications services, now branded as Switching for Business (SforB). Ofcom has not mandated a single technical solution for this process.
GPLB-SG	GPLB Steering Group	An independent body of industry stakeholders together with the OTA2, that guides and facilitates the steering meetings and publication of the SforB process documentation via FCS. The group retains the historical "GPLB" label even though the process has been renamed to Switching for Business (SforB).
HMAP	Hub Managed Access Provider	A specialised MAP type that operates as a hub-and-spoke delivery mechanism (e.g., TOTSCo). An HMAP accepts messages and forwards them among its subscribed members without engaging with the message payload.
JAM	JSON Asynchronous Message	The industry-wide messaging standard published on the OTA2 website. JAM defines the envelope format (addressing, routing, correlation) used by all CPs and MAPs regardless of the underlying transport.
MAP	Managed Access Provider	An organisation that facilitates message exchange on behalf of CPs by offering integration services, portals, and technical solutions. A CP can become a MAP of 1 (itself) by abiding by MAP rules.
MoU	Memorandum of Understanding	The agreement governing collaboration between MAPs and CPs participating in the CSF.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

ABBREVIATION	FULL TERM	DEFINITION
OTA2	Office of the Telecommunications Adjudicator	The industry body accountable for publishing and controlling RCPID Standards and the JSON Asynchronous Messaging Specification.
OTS	One Touch Switch	The Ofcom-mandated process for switching residential IAS and NBICS services between consumer CPs. All OTS messages must be routed via the TOTSCo Hub.
PKI	Public Key Infrastructure	A framework for managing cryptographic key pairs and digital certificates. In the CSF, PKI is implemented via DKIM (RFC 6376) to sign and verify messages between MAPs.
RCPID	Retail Communications Provider Identifier	A unique identifier assigned to a CP brand for the purpose of message exchange. In the CSF, RCPIDs use the UUIDv4 format (RFC 4122) and remain with the CP for life, even when changing MAPs.
RFC	Request for Comments	A publication series from organisations such as the IETF that defines internet standards, protocols, and technical specifications.
SforB	Switching for Business	The current branding for the GPLB process — the industry process for switching business telecommunications services.
TAG	Telecom Technical Architecture Group	A consortium of telecoms industry stakeholders responsible for the development, publication, and change control of the CSF. The TAG meets weekly via its steering group.
TLS	Transport Layer Security	A cryptographic protocol providing secure communication over networks. The CSF mandates TLS 1.3 (RFC 8446) for all connections.
TOTSCo	Telecoms One Touch Switching Company	The organisation operating the centralised hub for OTS message routing. TOTSCo also publishes OTS process and technical documentation. For SforB, TOTSCo may operate as an HMAP within the CSF.

Key Definitions

CP Registry

A JSON data structure published by each MAP containing detailed information about that MAP, its connected CPs, and their supported processes (including RCPIDs, brand names, PKI signing domains, resource URLs, and contact details). Both the MAP and each individual CP can include a `contact` array using the same structure — the MAP contact is for MAP-to-MAP technical issues, while each CP's contact is for CP-to-CP operational escalations (sales, support, technical). Each MAP publishes its own CP Registry and makes it available to other MAPs via an OAuth 2.0-protected API endpoint. The CP Registry is the outward-facing data that MAPs share with each other.

Master Registry

The private, consolidated view built by each MAP when it collects and converges all other MAPs' CP Registries. The Master Registry contains the full routing, PKI signing, and contact details for every CP across the entire CSF network. It is held privately by the MAP and used

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

internally for message routing, DKIM signature verification, and CP-to-MAP resolution. Each MAP maintains its own copy of the Master Registry.

Directory

A filtered, cut-down version of the Master Registry that a MAP shares with its own CPs. The Directory lists the brand names and RCPIDs available to switch with — the information CPs need to populate drop-down lists, search-as-you-type fields, and message addressing. It excludes inter-MAP routing details, PKI signing domains, and other sensitive operational information. When a CP selects a destination from the Directory, the MAP uses the RCPID to look up the full routing and signing details in its private Master Registry.

Sponsor MAP

An onboarded MAP that is assigned (via round-robin rotation) to guide a new MAP through the onboarding process. The Sponsor MAP validates the new MAP's technical readiness, conducts testing, and acts as a gatekeeper to the CSF network.

New MAP

An organisation that wishes to join the CSF and begin exchanging messages with other MAPs. A New MAP must complete the sponsored onboarding process before it can publish its registry or exchange production messages.

Bit pipe

Refers to a network provider that simply moves data without adding value or examining content. The CSF is, by design, a bit pipe for SforB and any other industry processes that use it — it carries the JAM envelope and does not interpret the message body. The full normative definition is in Definitions §2.3.

SOR (Switch Order Reference)

A reference number associated with a Switch Order. SORs have a validity period (typically 31 days) and cannot be transferred between MAPs; they must be re-issued. The full definition is in Definitions §2.6.

Letterbox

The REST API endpoint used by MAPs to send and receive messages. The letterbox accepts JAM-formatted JSON messages over HTTPS (TLS 1.3) with OAuth 2.0 authentication and returns an HTTP 202 to acknowledge receipt.

Routing Group

A named collection of routing ID patterns (expressed as regular expressions) that determines which message types are accepted by a specific MAP service endpoint. For example, `businessSwitch.*` for SforB messages or `residentialSwitch.*` for OTS messages.

Feature Release

A mechanism within the CSF that allows MAPs to adopt new capabilities incrementally without requiring all MAPs to upgrade simultaneously. MAPs advertise their supported

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

feature sets in the registry, enabling automatic negotiation of the highest compatible version between interacting parties.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

PART 2

Implementation

Technical reference — for developers, integration engineers, and technical architects

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

1. Getting Started

1.1 Purpose

This section provides a practical checklist for MAPs implementing the CSF. Whether you are a new MAP joining the network or an existing MAP adding CSF support, this guide covers the prerequisites and steps to get operational.

1.2 Prerequisites

Before beginning implementation, ensure you have:

- [ ] Read Part 1: Framework to understand the architecture and principles
- [ ] Reviewed the JSON Asynchronous Messaging Specification (JAM) published on the OTA2 website
- [ ] Reviewed the **RCPID Standards** (in the process of being published by OTA2; no direct link available at the time of this publication — check the OTA2 website for the latest)
- [ ] Reviewed the Switching for Business Process Documents published on the FCS website
- [ ] Reviewed the TOTSCo Hub API Specification v2.0 for hub interoperability requirements
- [ ] Identified a Sponsor MAP (or been assigned one via the rotation)
- [ ] Agreed terms and costs with the Sponsor MAP for the onboarding process

1.3 Implementation Checklist

Infrastructure Setup

- [ ] Provision TLS 1.3-capable endpoints for:
- Letterbox API (message sending/receiving)
- Directory API (registry publishing)
- OAuth 2.0 token endpoint
- [ ] Configure OAuth 2.0 authorisation server (Client Credentials flow)
- [ ] Set up PKI key management (generation, storage, rotation)
- [ ] Implement DNS management for CP `_domainkey` and `_mapkey` TXT records

API Implementation

- [ ] Implement the **Letterbox API** — accepting JAM-formatted JSON messages via HTTP POST, returning HTTP 202 on success (see Message API)

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- [ ] Implement the **Directory API** — serving your registry as a JSON document via HTTP GET (see Directory API)
- [ ] Implement **OAuth 2.0 token endpoint** — issuing access tokens to authenticated MAPs (see OAuth 2.0 Implementation)

Security Implementation

- [ ] Implement **DKIM message signing** for outbound messages (see PKI & DKIM Signing)
- [ ] Implement **DKIM signature verification** for inbound messages
- [ ] Implement **DNS-based CP verification** — validating the `_domainkey` and `_mapkey` records
- [ ] Generate PKI key pairs for each CP (minimum RSA-2048 or Ed25519)

Directory and Registry

- [ ] Build and publish your **MAP registry** in the specified JSON format
- [ ] Implement **registry collection** — pulling other MAPs' registries via their directory API
- [ ] Implement **directory convergence** — merging collected registries into a unified directory
- [ ] Implement **conflict resolution** — handling CPs that appear in multiple registries

Message Processing

- [ ] Implement **message validation** — envelope structure, RCPID verification, signing validation
- [ ] Implement **retry logic** with exponential backoff for failed deliveries
- [ ] Implement **asynchronous error handling** using 9xxx (industry) and 8xxx (CSF) error codes
- [ ] Implement **message delivery failure notifications** to the originating CP/MAP

Operations

- [ ] Set up **monitoring and alerting** for endpoint availability
- [ ] Implement **status reporting** in the registry (incident flags, planned changes)
- [ ] Establish **audit logging** for all message exchanges
- [ ] Implement **rate limiting** and denial-of-service protections

Onboarding

- [ ] Complete **Sponsor MAP assessment** — technical, security, and compliance review

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- [ ] Establish **OAuth 2.0 credentials** with the Sponsor MAP
- [ ] Pass **connectivity testing** with provisional/restricted access
- [ ] Pass **process-level testing** with test CPs
- [ ] Pass **operational validation** — delivery reliability, retry, and SLA compliance
- [ ] Pass **exit criteria** — including hidden unit tests and PKI verification
- [ ] Establish **OAuth 2.0 credentials** with all other MAPs
- [ ] Obtain **full access** approval from Sponsor MAP

TOTSCo Integration (if applicable to the protocol e.g. SforB)

- [ ] Register CSF endpoints with TOTSCo using existing procedures
- [ ] Configure a **dedicated endpoint for TOTSCo** (separate from CSF MAP endpoint)
- [ ] Ensure TOTSCo can access your **Directory API** using existing OAuth 2.0 credentials
- [ ] Test message exchange with TOTSCo using their existing letterbox interface

> NOTE
> See TOTSCo Integration for detailed integration guidance.

1.4 Recommended Endpoint Structure

While the CSF does not mandate specific URL patterns, the following structure is recommended:

ENDPOINT	URL PATTERN	PURPOSE
OAuth Token	`https://{fqdn}/oauth2/token`	Issue access tokens
Directory API	`https://{fqdn}/directory`	Serve MAP registry
Letterbox (CSF)	`https://{fqdn}/letterbox/csf/{version}/post`	Receive messages from CSF MAPs
Letterbox (Hub)	`https://{fqdn}/letterbox/hub/{version}/post`	Receive messages from HMAPs (e.g., TOTSCo)
Service Status	`https://{fqdn}/status`	Operational status page/API

> BEST PRACTICE
> : Maintain separate letterbox endpoints for CSF MAPs and HMAPs (e.g., TOTSCo). This allows the TAG to iterate the CSF rapidly without affecting established hub exchange networks.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

1.5 Development Resources

RESOURCE	LOCATION
PKI verification testing tool	CSF Documentation Wiki
CSF test endpoints	Available from TAG members for integration testing
Virtual Test Host	Provided by MAPs for local onboarding testing
Sample code (Java, C#)	See PKI & DKIM Signing — Worked Examples
OpenAPI specification	See `letterbox-openapi.yaml` in the CSF documentation repository
TOTSCo Basic Connectivity testing	https://totsco.org.uk/business-user-guide/

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

2. OAuth 2.0 Implementation

2.1 Overview

The CSF uses **OAuth 2.0 Client Credentials flow** for machine-to-machine authentication between MAPs. This is the same approach used by TOTSCo for OTS, ensuring maximum compatibility across the industry.

Each MAP operates as both:

- An **OAuth 2.0 Authorisation Server** — issuing tokens to other MAPs that want to access its endpoints
- An **OAuth 2.0 Client** — requesting tokens from other MAPs to access their endpoints

During onboarding, MAPs exchange `client_id` and `client_secret` credentials with each other.

2.2 Access Token Request

To obtain an access token, the requesting MAP sends an HTTP POST to the target MAP's token endpoint.

Request URL: `https://{fqdn}/oauth2/token`

The FQDN is obtained from the `oauthTokenURL` field in the target MAP's registry (see Directory API).

Request Headers

HEADER	VALUE	NOTES
`Authorization`	`Basic <Base64(client_id:client_secret)>`	Base64-encoded credentials provided during onboarding
`Content-Type`	`application/x-www-form-urlencoded`	Standard OAuth 2.0 form encoding

Request Body Parameters

FIELD	TYPE	REQUIRED	DESCRIPTION
`grant_type`	String	yes	always `client_credentials`
`client_id`	String	Yes	Client identifier provided by the target MAP during onboarding
`client_secret`	String	Yes	Client secret provided by the target MAP during onboarding

Note : A MAP should support `client_id` + `client_secret` in both headers and body as per OAuth 2.0 Standard

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

2.3 Access Token Response

A successful request returns an HTTP 200 with the following JSON body:

```
{
  "access_token": "{OAuth2 Access Token}",
  "token_type": "bearer",
  "scope": "default",
  "expires_in": 3600
}
```

FIELD	TYPE	MANDATORY	DEFAULT	DESCRIPTION
`access_token`	String	Y	—	The bearer token to include in subsequent API requests
`token_type`	String	Y	`bearer`	The type of token issued
`scope`	String	N	`default`	The scope of access granted
`expires_in`	Integer	Y	`3600`	Token validity in seconds (typically 1 hour)

2.4 Using the Access Token

Include the access token in the `Authorization` header of all API requests:

```
Authorization: Bearer {access_token}
```

The token MUST be validated by the receiving MAP on every request, checking:

- Token signature validity
- Token has not expired
- Issuer matches the expected authorisation server
- Audience/scope is appropriate for the requested resource

2.5 Token Lifecycle

- Tokens are **short-lived** (typically 1 hour)
- MAPs SHOULD cache tokens and reuse them until close to expiry
- Client secrets MUST be stored securely (e.g., in environment variables or secret vaults, never in source code)
- Client secrets SHOULD be rotated periodically

2.6 Error Responses

The following errors may be returned during token generation:

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

HTTP CODE	ERROR	ROOT CAUSE
400	Bad Request	Invalid request parameters or missing required fields
401	Unauthorised	Incorrect client credentials (`client_id` or `client_secret`)
404	Not Found	Incorrect token generation URL
405	Method Not Allowed	Incorrect HTTP method (must be POST)
415	Unsupported Media Type	Missing or incorrect `Content-Type` header

2.7 Security Requirements

REQUIREMENT	MANDATORY
All token exchanges over TLS 1.3	Yes
Validate token signature, issuer, and audience on every request	Yes
Limit token scope and lifetime	Yes
Never expose client secrets in frontend code or logs	Yes
Store client secrets in secure vaults or environment variables	Yes
Rotate client secrets periodically	Recommended
Use Proof Key for Code Exchange (PKCE) for public clients	If applicable

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

3. PKI and DKIM Message Signing

3.1 Overview

Without a centralised authentication body, the recipient of a message must verify that:

1. The originator is who they claim to be
2. The MAP is authorised to act on behalf of the CP
3. The message has not been altered in transit

The CSF achieves this using **DKIM** DomainKeys Identified Mail, RFC 6376 — a well-established PKI-based signing standard originally designed for email, adapted here for HTTP message signing between MAPs.

DKIM is ideally suited to a federated network where trust has been established through onboarding, but ongoing verification of the CP/MAP relationship and payload integrity is required.

3.2 Public/Private Key Pairs

The CSF uses asymmetric cryptography. Each CP has a **key pair**:

- **Private key**: Held securely by the MAP. Used to sign outbound messages on behalf of the CP.
- **Public key**: Published in the CP's DNS TXT records. Used by receiving MAPs to verify signatures.

If the signature encrypted with a private key can be correctly decrypted with the corresponding public key, the recipient knows the signed payload originated from the holder of that private key.

3.3 DNS Records

Every CP registered with a MAP maintains **two DNS TXT records**:

Domain Key Record

```
[RCPID]._domainkey.[cp-domain]
```

Contains the public key in RFC 6376 format. Example:

```
7d87c2a7-4f64-433c-8ae1-9460f19a0a13._domainkey.supercompany.com
```

TXT value: `k=rsa; p=MIIBojANBgkqhkiG9w0BAQE...` (Base64-encoded public key)

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

MAP Key Record

```
[RCPID]._mapkey.[cp-domain]
```

Contains the URL of the MAP's message endpoint. Example:

```
7d87c2a7-4f64-433c-8ae1-9460f19a0a13._mapkey.supercompany.com
```

TXT value: `https://switching.supermap.com/api/`

Why Two Records?

- The `_domainkey` record enables signature verification by any receiving MAP
- The `_mapkey` record provides routing verification — the MAP name from the directory could be spoofed, but the DNS record cannot
- Having the RCPID in the domain allows a CP to have **multiple RCPIDs hosted at different MAPs** if required

CP Responsibilities

The CP only needs to create/update these two DNS records. All remaining processing is performed by the MAPs. During onboarding, the MAP (or CP) generates the key pair, places the public key in the `_domainkey` record, and gives the private key to the MAP.

MAP Responsibilities

MAPs MUST NOT cache a CP's public key in-process beyond the published DNS TTL — a stale public key honoured after rotation creates exactly the post-revocation forgery risk DKIM/PKI is designed to prevent.

To ensure that the DNS records used to verify CSF messages reflect the live public records as closely as possible, **MAPs SHOULD operate a local DNS server** that mirrors the public DNS entries for `_domainkey` and `_mapkey` records in near-real-time, observing the public TTLs published by each CP. The core standard for locally served DNS zones is **[RFC 6303 — Locally Served DNS Zones](https://datatracker.ietf.org/doc/html/rfc6303)**.

This is a common operational pattern — for example, an authoritative-only recursive resolver pinned to upstream authoritative servers, or a stub resolver bypassing the OS resolver cache. It is widely deployed software and does **not** impose new infrastructure cost on MAPs.

Why a local DNS server matters in CSF:

- **Reduced latency.** DNS lookups become near-local rather than going through general internet resolver chains, lowering per-message verification time.
- **Live-or-near-live freshness.** The MAP sees public DNS changes as soon as published TTLs allow, rather than being subject to upstream resolver caches that may serve stale records past the published TTL.
- **No missed CP key rotations or MAP changes.** When a CP rotates its `_domainkey` or updates its `_mapkey` (e.g. during a MAP change), the MAP picks up the new value at TTL expiry without depending on the behaviour of public recursive resolvers.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Impact if not implemented. A MAP that relies on the default OS resolver cache (or any other shared recursive resolver) may:

- **Continue to honour stale `_domainkey` records** — accepting signatures verified against a CP's previous public key after the CP has rotated. This is the primary attack surface for post-revocation key forgery and is **the single largest reason** the local-DNS-server recommendation matters.
- **Route via a stale `_mapkey`** — sending messages or registering counterparties at the CP's previous MAP after the CP has moved. The destination MAP's `9301` reject path eventually corrects this, but only at the cost of delivery latency and false-failure noise on inter-MAP telemetry.
- **Misdiagnose its own DNS-related `8102 TEMP_FAIL` rejections** — opaque upstream-resolver behaviour can make transient DNS lookups look like systemic issues, hampering operational triage and slowing incident response.

MAPs that choose not to operate a local DNS server SHOULD nevertheless document the operational implications they have accepted (the cache TTL of their chosen recursive resolver, their monitoring of stale-data incidents) and SHOULD treat the resulting freshness gap as a known operational risk in their MAP MoU with CPs.

3.4 DKIM Signature Process

The signing process follows RFC 6376:

```
body-hash  = hash-alg(canon-body)
data-hash  = hash-alg(h-headers, D-SIG, body-hash)
signature  = sig-alg(d-domain, selector, data-hash)
```

Where:

TERM	DESCRIPTION
`body-hash`	Hash of the message body using the specified algorithm
`hash-alg`	The hashing algorithm (from the `a` parameter)
`canon-body`	Canonicalised body using the algorithm in the `c` parameter
`data-hash`	Hash of headers + DKIM-Signature header + body hash
`h-headers`	Headers listed in the `h` parameter, to be included in the signature
`D-SIG`	The DKIM-Signature header without the `b` value
`signature`	The final signature value
`sig-alg`	The signing algorithm (from the `a` parameter)
`d-domain`	The domain from the `d` parameter
`selector`	The selector from the `s` parameter (always the CP's RCPID)

> NOTE
> : The `l` parameter from RFC 6376 has been removed. While helpful for large emails, it is not relevant for CSF messages that are typically a few hundred KB at most.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

3.5 CSF HTTP Headers

The CSF uses two HTTP headers for message signing:

HEADER	PURPOSE	FORMAT
`X-CSF-SIGNATURE-DATESTAMP`	Timestamp of signature creation	`yyyyMMddHHmmssS` (S = tenth of a second)
`X-CSF-SIGNATURE`	The DKIM signature	See structure below

X-CSF-SIGNATURE Structure

Example:

```
a=rsa-sha256; c=simple/simple; s=809b6e65-a6e7-40f6-8b52-04dd65b6fce1;
d=gplb-test.nowyoyo.net; bh=/CWGLVBT2gdyO5cudfAEUSF43KHLmlJjU/Nr2YNqkos=;
h=X-CSF-SIGNATURE-DATESTAMP;
b=4J4OGZfxiIetZP8WMMt56+MBqX0W/KdV9pAxzeS4PBC5rjmRtQc9PNMfkd0+...
```

Mandatory Tags

TAG	DESCRIPTION
`a`	Algorithms used for hashing and encryption (e.g., `rsa-sha256`)
`s`	Selector — always the CP's RCPID
`c`	Canonicalisation algorithms — always `simple/simple` for CSF HTTP messages
`d`	Domain for DNS public key lookup
`h`	Header fields included in the signature (must include `X-CSF-SIGNATURE-DATESTAMP`)
`bh`	Base64 hash of the message body
`b`	Base64 signature data

Mandatory Tags with Fixed Values

TAG	VALUE	DESCRIPTION
`v`	`1`	DKIM version

Optional Tags with Fixed Values

TAG	VALUE	DESCRIPTION
`q`	`dns/txt`	Method of acquiring the public key

DKIM-Signature Canonicalisation

During signature creation and verification, the DKIM-Signature MUST be formatted as:

```
DKIM-Signature: <signature value>
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

> IMPORTANT
> : Note the exact casing, the colon immediately after the header name, and the single space (ASCII 0x20) after the colon. Any variance will cause valid signatures to fail verification.

3.6 Supported Algorithms

Encryption and Signing

ALGORITHM	HASH	`a` TAG VALUE
RSA	SHA2-256	`rsa-sha256`
RSA	SHA2-512	`rsa-sha512`
Ed25519	SHA2-256	`ed25519-sha256`
Ed25519	SHA2-512	`ed25519-sha512`

> NOTE
> : SHA-1 has been removed as it is no longer secure. SHA-3 is not used.

Key Length Requirements

- **RSA**: Minimum **2048-bit** keys. Verifiers SHOULD reject anything smaller. Avoid RSA-4096 for DKIM unless mandated, due to higher CPU cost and potential DNS/UDP size issues.
- **Ed25519**: Fixed **256-bit** keys only (no size choice). Ed25519 keys are significantly shorter than RSA keys.

DNS Key Examples

RSA 2048-bit public key TXT record:

```
k=rsa; p=MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA6/RekuGAsET2Ir8E4K5BU74MBXe
bo7QwrOnSGirtNgRLDcwP2R84Eeum9BLxZtiumUrmO6ja7DrteSZhzOyC09KM3b22zV43bm7nv/FF40+
KpTDQ/FZ32W838UIZGgHOyICk37hlPvnGrVPg7rT6HAKsaj5W0zbvO5vKDUOz2r3bvrWo79YSxtVfGQ1
pHhhZYV2dL1C0uiADbz5L1Cu0qCBjqGJQbYq4ALdaKTnAm2NG/N9W2i3IWJUQIyiTr11qtBrrR8sj2Yh
Yeph6GTsOXggpAWZnEHZz+LKzD4roVqUxkByLWRLB2I5Ju/xPPtTjCDakfvGlsiWl5BMAhn79p0mW6hz
cA4DT7Iz+nSV8K3XMU4rxs0AcaafcBsKREmfbGz4K3ycWHZ/Yrxb+PqYHy7NSK76mHsAjuU8nuKYtZGI
YHwT4friS2R3HtXJ+olLmdZ+XnkpdvEGDb+ZVv/G9vYp020LZQBBd0hp1Ez8a/3F7jRt8xUJ0ljstWjb
q+m3bAgMBAAE=
```

Ed25519 public key TXT record:

```
k=ed25519; p=MCowBQYDK2VwAyEAqiCrokPReIFI1h4Jdp7RCRRVQ4Fltdyn429O4H2jYog=
```

Canonicalisation

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

The CSF uses `simple/simple` canonicalisation for both headers and body. This is the most straightforward implementation — no post-processing of headers or body content is required.

3.7 Public Key Storage

The public key is stored as a DNS TXT record following DKIM standards.

Domain format: `[s]._domainkey.[d]`

Where `s` and `d` are the values from the `X-CSF-SIGNATURE` header. The `s` value is always the CP's RCPID.

Example: `809b6e65-a6e7-40f6-8b52-04dd65b6fce1._domainkey.gplb-test.nowyoyo.net`

The CP's RCPID MUST also match an entry in the directory, which defines the valid domain (`d`) for that CP's public key. Any request where `d` does not match the directory entry MUST be rejected.

3.8 Worked Examples

What follows are simplistic Low-Level worked examples of the DKIM process for signing and verifying. However, it is possible that developers are able to make use of Higher-Level libraries that already implement the DKIM RFC and remove a lot of the low level implementation. See Higher Level Libraries

3.8.1 Java — Creating a Signature

```
public String createSignature() throws Exception {
  byte[] httpBody = """
      {
        "test_field": "Test Data"
      }""".getBytes();


  String privateKeyPEM =
"MIIG/gIBADANBgkqhkiG9w0BAQEFAASCBugwggbkAgEAAoIBgQDr9F6S4YCwRPYivwTgrkFTvgwFd5ujtDCs6dIaKu02BEsNzA/ZHzgR66b0EvFm2K6ZSuY7
qNrsOu15JmHM7ILT0ozdvbbNXjdubue/8UXjT4qlMND8VnfZbzfxQhkaAc7IgKTfuGU++catU+DutPocAqxqPlbTNu87m8oNQ7Pavdu+tajv1hLG1V8ZDWkeG
FlhXZ0vULS6IANvPkvUK7SoIGOoYlBtirgAt1opOcCbY0b831baLchY1RAjKJOvXWq0GutHyyPZiFh6mHoZOw5eCCkBZmcQdnP4srMPiuhWpTGQHItZEsHYjk
m7/E8+1OMINqR+8aWyJaXkEwCGfv2nSZbqHNwDgNPsjP6dJXwrdcxTivGzQBxpp9wGwpESZ9sbPgrfJxYdn9ivFv4+pgfLs1IrvqYewCO5Tye4pi1kYhgfBPh
+uJLZHce1cn6iUuZ1n5eeSl28QYNv5lW/8b29inTbQtlAEF3SGnUTPxr/cXuNG3zFQnSWOy1aNur6bdsCAwEAAQKCAYANK3PJUfZpDfqbSXIACUVJdcomsfgA
AFfpK2HYFH0lZXk7z/tXU2yvP2C4il79G+soB1nVr1exXnaimPMxNtliOt99yljr42UjFJroGlxSwL3e+TefyowE9Fq5EOub5Mk490kzazRMDJ7N4EQBPtOyE
0D7ofb5qOGlCU2lRXV5Fd1Uzy1RgsrWN9K+Xd4KWVbbObznmWMXRLIFhd4dNHneGnSzPxavsxrsNr1hVmXNXVuLVB+h0XMS5PHkbHre5vOWwk0x/BpqHtA15s
U6tdtWAem1zR8cKDn8D4uVSIU91K6sAqSR34YpJAb79GiEtTFdXCA1IPMyBRHmwF2jiK0UZpbBhuYnLkwcbSMul7xaMlEHMTxZxgMFK765IKtK/cQjENMCcFO
9JSRkTMRsnD8pjNU846keyMFUZ1ap2FddUpfNuAV2f9844QNJyF07oly4gmx/+ruzjWjWN+ByUJvB7zEZh/zHcUCgd9+Ixay1F/AD6Dub3mIh/toi2VqOFnkC
gcEA7TM/oQu8udZc3M9IWdipJV2iehhPqXAIUqipQnICK780qYRW6UdVSXk1twEAg+VymITOht1ICCzyqKP9KhNLRKfAnZLJkV1zEEJin19gp6YNoPWfaJEwU
rpb6WjgvADdPuq+teZ+Hk5G8MoOKkqDt6rlrpdtib1mHTogyKKfjClX+cmwU2AkkcYWm/T/7tO3+aPWWFN+B70PUEf/K5OEfKIPiSGGPg04DWki99M+d+Gnt9
OzrbDxC985MkZT43wZAoHBAP6n2PELun2BD2rEqUDV5HBDAewKSYWsyOsfTbvLMa1ALu54h/FQsrPKkgnkpPBGKRaDtgSixcigVvHKb6jm96Zn43pg7uxCq9Q
pKfg9pUj3orfNvw7GmgEg/J8vW8T6ux3jlhOsY12XegmabOvYHr0/M6zvX1KWtLXig2SZN53eAFCYMbkBrcwnDjgIcai48wasapa0p8mD3xoBOG14+lL2Y/DF
HgoF+2l3njISZGBntBc/tgyOa104TqMy8+D4EwKBwQDHpk02RmTRnsaG7MmfJigo1Uk+r1vN6Ah5WpEs5j1BiSzQSh3FOE9nCmjV4jgGzIfKLG6RQYuxpfORU
oZydc7yuKf9eWHDwv5ofxf3wRXfxnrOMi+8mggsecOHEMmoNKoEnR1sidc5tvUrE0cc/Z8kZunwLHD8cLiUfSq+9XKJTPtJuiN56gCd2jeJiYwp/3Zo3yg5K/
12kgFjt1Xl3cK0DMw6xkbxz7qQPyA5rEp2KS88ISqpVbduILNJx7wwS3ECgcEAzQkZ3CLEWc6zOhTz7bcKAfWBs6oovk97SgxfSyf0bHk0EF/NnNeLusUMRpj
o0Gi9JlqQEDV6p+mpd2617rlghoQ5HMy1MlcQAHfQSgZgcVqpkfI/tcbkMqp7nDPGYNg8Fnmq2VZAfxe6c8b5kf7l6RvdII1vI5EiGRwzDKlspVgcysdvqXUX
mTuM8EKkOOQJEMN74rG8Mr1RwZ9f7oysiGXH3BDp+coNPkLIhapXVWPKFbn/eyakfV8bub0JrYYvAoHAc0forosmO0gtkBXOo4XJlEd4zZ7IoqEHoVarIodtI
lSBDtEXibvKvuEoSlSosvskaUcc2nk3ZJPx2VkRkPt0lad9IY0bQde7mcpiYR4itCaj4Siz0JyKryPBL2lC/PlgkSpLCK92C8MEFrZOhXYnvrng2pP4EWI2s6
xIrxi62s2OMyAkXoLcOI4pZKztDRwVs1M+vHuaNQtaRPcA8cf0MUQ5KbcXUixwhbfggN7hASQGYz1+AYaowtDBZATPS6QQ";
  KeyFactory keyFactory = KeyFactory.getInstance("RSA");
  PrivateKey privateKey = keyFactory.generatePrivate(new PKCS8EncodedKeySpec(Base64.getDecoder().decode(privateKeyPEM)));


  MessageDigest digest = MessageDigest.getInstance("SHA256");
  // Get the body hash, using simple canonicalization (2nd part of 'c') as per
https://datatracker.ietf.org/doc/html/rfc6376#section-3.4.3
  String bh = Base64.getEncoder().encodeToString(digest.digest(httpBody));
  // bh = IVIj2cQQOAapFmSJl6X0y6dQgKWhYHqQetWe9mWINNQ=


  // Assumes we are using a CP with ID 809b6e65-a6e7-40f6-8b52-04dd65b6fce1 that is hosted on the MAP Nowyoyo.
  String sigTemplate = "a=rsa-sha256; q=dns/txt; c=simple/simple; s=809b6e65-a6e7-40f6-8b52-04dd65b6fce1; d=gplb-
test.nowyoyo.net; v=1; h=X-CSF-SIGNATURE-DATESTAMP; bh=" + bh + ";";
  // sigTemplate = a=rsa-sha256; q=dns/txt; c=simple/simple; s=809b6e65-a6e7-40f6-8b52-04dd65b6fce1; d=gplb-
test.nowyoyo.net; v=1; h=X-CSF-SIGNATURE-DATESTAMP; bh=IVIj2cQQOAapFmSJl6X0y6dQgKWhYHqQetWe9mWINNQ=;
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
  // Normally we would parse the 'a' tag to get the correct algorithm, but this is a simple example.
  Signature signature = Signature.getInstance("SHA256withRSA");
  // Initialise with the Private Key
  signature.initSign(privateKey);
  // As we are using simple canonicalization (1st part of 'c'), we add the header value only
  signature.update("202412121340391".getBytes());
  // Each Header must be appended with \r\n as per https://datatracker.ietf.org/doc/html/rfc6376#section-3.7
  signature.update("\r\n".getBytes());
  // Append the signature using simple canonicalization (1st part of 'c') as per
https://datatracker.ietf.org/doc/html/rfc6376#section-5.6
  signature.update(("DKIM-Signature: " + sigTemplate+ " b=").getBytes());


  // append the signature to the sigTemplate
  String finalSignature = sigTemplate + " b=" + Base64.getEncoder().encodeToString(signature.sign());
  // finalSignature = a=rsa-sha256; q=dns/txt; c=simple/simple; s=809b6e65-a6e7-40f6-8b52-04dd65b6fce1; d=gplb-
test.nowyoyo.net; v=1; h=X-CSF-SIGNATURE-DATESTAMP; bh=IVIj2cQQOAapFmSJl6X0y6dQgKWhYHqQetWe9mWINNQ=;
b=rSnlux6S7feBZBEA3wl0X+XA6RfNq8MFnS2nMo0dgEMNmQguNhRIpzuPBTU/BgPmZA57IXF4eMN886rFC5Lo0yy5cuSZpHaVbI5gvf/1hX/SAMJfa2PAipn
1/9+yeUj8ajnUYMsO6v1i6bx6LA5Gq+mJEBm4XRwA3Ps3wdgJ1bzTQ7s1L8+/iXci1OqieMdWHvdBi7uncBhpFUQdHVEYZ7zpzDjG59B2i1Z2BGFBEfcTF9FJ
PfJbJoabUuzT1iZlcW1s18tnZyav/4X3ZTM5YIVGYo355gtiT4YNEmz+mwFN/cOvRkE+X7ukFvNrXOfjNTtfBf2aGWOxMuSFX9UkHi74Vyi5/Snim9Qp1OrKC
YRhb+6NaqXev4rbs244t7NTAy0D3WcWxWZN5tf6R8rQ1jm1LklaX2xtr7GaM07mPGRFvGGukT7VSzo1nxD9JDKTiP/C8cA5+kunIFXNrst6xXI+0GYC3QiPZy
msbsYEcp06AyOH/QZjUoGO77lAbZKJ


  // The result is then added to the HTTP Request as Header X-CSF-SIGNATURE
  return finalSignature;
}
```

3.8.2 Java — Verifying a Signature

In this code section, it is assumed that the Public Key has already been acquired from the DNS entry in the signature and has been verified that it matches the details from the directory

```
  public boolean verifySignature(PublicKey publicKey, HttpHeaders httpHeaders, byte[] httpBody) throws
NoSuchAlgorithmException, InvalidKeyException, SignatureException {
  // Get Signature and Timestamp from header
  String signatureHeader = httpHeaders.getFirst("X-CSF-SIGNATURE");
  // pull out the headers stated in the 'h' tag, they are appended to the signature in the order they appear in 'h'
  // in this simplistic example we already know that there was only 1 field mentioned.
  String timestampHeader = httpHeaders.getFirst("X-CSF-SIGNATURE-DATESTAMP");
  // timestampHeader = 202412121340391


  // get the Base 64 decoded value of the 'b' tag
  byte[] signatureBytes = getSignatureBytes(signatureHeader);
  // get the signature line without the 'b' tag
  String signatureTemplate = unsignedSignature(signatureHeader);
  // signatureTemplate = a=rsa-sha256; q=dns/txt; c=simple/simple; s=809b6e65-a6e7-40f6-8b52-04dd65b6fce1; d=gplb-
test.nowyoyo.net; v=1; h=X-CSF-SIGNATURE-DATESTAMP; bh=IVIj2cQQOAapFmSJl6X0y6dQgKWhYHqQetWe9mWINNQ=;


  MessageDigest digest = MessageDigest.getInstance("SHA256");
  // Get the body hash, using simple canonicalization (2nd part of 'c') as per
https://datatracker.ietf.org/doc/html/rfc6376#section-3.4.3
  String bh = Base64.getEncoder().encodeToString(digest.digest(httpBody));
  // bh = IVIj2cQQOAapFmSJl6X0y6dQgKWhYHqQetWe9mWINNQ=


  // Check the bh is the same as the bh value in the signature. throw exception if not
  verifyBodyHash(signatureHeader, bh);


  // Normally we would parse the 'a' tag to get the correct algorithm, but this is a simple example.
  Signature signature = Signature.getInstance("SHA256withRSA");
  // Initialise with the public Key for Verification
  signature.initVerify(publicKey);
  // As we are using simple canonicalization (1st part of 'c'), we add the header value only
  signature.update(timestampHeader.getBytes());
  // Each Header must be appended with \r\n as per https://datatracker.ietf.org/doc/html/rfc6376#section-3.7
  signature.update("\r\n".getBytes());



  // Append the signature using simple canonicalization (1st part of 'c') as per
https://datatracker.ietf.org/doc/html/rfc6376#section-5.6
  signature.update(("DKIM-Signature: " + signatureTemplate + " b=").getBytes());


  // simple true/ false result
  return signature.verify(signatureBytes);
}
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

3.8.3 C# — Complete Sign and Verify Example

```
using Org.BouncyCastle.Crypto;
using Org.BouncyCastle.Security;
using System.Security.Cryptography;


// === SIGNING ===


string csfTimestamp = DateTime.Now.ToString("yyyyMMddHHmmssf");
string cpId = "809b6e65-a6e7-40f6-8b52-04dd65b6fce1";
string hValue = "X-CSF-SIGNATURE-DATESTAMP";


// Parse private key
byte[] privateKeyBytes = Convert.FromBase64String(privateKeyPEM);
AsymmetricKeyParameter privateKey = PrivateKeyFactory.CreateKey(privateKeyBytes);


// Hash the message body
SHA256 sha256 = SHA256.Create();
byte[] messageBytes = System.Text.Encoding.ASCII.GetBytes(GetMessage());
byte[] messageBodyHashBytes = sha256.ComputeHash(messageBytes);
string bh = Convert.ToBase64String(messageBodyHashBytes);


// Build signature template
string signature = $"a=rsa-sha256; q=dns/txt; c=simple/simple; "
    + $"s={cpId}; d=gplb.onetouchgateway.net; v=1; "
    + $"h={hValue}; bh={bh};";


// Organise data to sign
string dataToSign = csfTimestamp + "\r\n" + "DKIM-Signature: " + signature + " b=";


// Sign with RSA private key
ISigner signer = SignerUtilities.GetSigner("SHA256withRSA");
signer.Init(true, privateKey);
signer.BlockUpdate(
    System.Text.Encoding.ASCII.GetBytes(dataToSign), 0, dataToSign.Length
);
byte[] signatureBytes = signer.GenerateSignature();


string csfSignature = signature + " b=" + Convert.ToBase64String(signatureBytes);


// === VERIFICATION ===


// Parse public key
byte[] publicKeyBytes = Convert.FromBase64String(publicKeyString);
AsymmetricKeyParameter publicKey = PublicKeyFactory.CreateKey(publicKeyBytes);


// Get unsigned portion
string unsignedSignature = csfSignature.Substring(0, csfSignature.IndexOf(" b="));


// Verify body hash
byte[] verifyBytes = System.Text.Encoding.ASCII.GetBytes(GetMessage());
string bhVerify = Convert.ToBase64String(sha256.ComputeHash(verifyBytes));
string bhInSig = /* extract bh from signature */;
if (bhInSig != bhVerify) throw new Exception("Hash mismatch!");


// Verify signature
string dataToVerify = csfTimestamp + "\r\n"
    + "DKIM-Signature: " + unsignedSignature + " b=";
ISigner verifier = SignerUtilities.GetSigner("SHA256withRSA");
verifier.Init(false, publicKey);
verifier.BlockUpdate(
    System.Text.Encoding.ASCII.GetBytes(dataToVerify), 0, dataToVerify.Length
);
byte[] sigBytes = Convert.FromBase64String(
    csfSignature.Substring(csfSignature.IndexOf(" b=") + 3)
);
bool isVerified = verifier.VerifySignature(sigBytes);
```

3.8.4 Higher-Level Libraries

Developers can use existing DKIM libraries with minor modifications, removing the need for low-level implementation:

- **Java**: Apache James jDKIM

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- **.NET**: MailKit

3.9 Error Codes

If a message fails signature verification, the receiving MAP MUST return one of the following error codes in a synchronous response body with **HTTP 403** status:

ERROR CODE	MEANING	RETRY?
`8101`	**PERM_FAIL** — The message failed verification. Retrying will not resolve the issue.	No
`8102`	**TEMP_FAIL** — The message failed verification due to a temporary problem (e.g., DNS failure).	Yes (with backoff)

> NOTE
> An `8102` response suggests a retry may resolve the issue, but does not guarantee it. The CP/MAP MUST have controls in place to handle delivery exceptions. Further details are in the CSF message delivery policy.

Response Format

```
{
  "errorText": "<description of the failure>",
  "errorCode": 8101
}
```

Example Responses

PERM_FAIL — Invalid domain for CP:

```
{
  "errorText": "Domain gplb-test.nowyoyo.net is not valid key source for CP 809b6e65-a6e7-40f6-8b52-04dd65b6fce1",
  "errorCode": 8101
}
```

PERM_FAIL — Body hash mismatch:

```
{
  "errorText": "Body Hash 3hzlA9zA8SAVapmd5ZZnMdwaZpk/WOyQXixrqVu/WMc= is different to signature
/CWGLVBT2gdyO5cudfAEUSF43KHLmlJjU/Nr2YNqkos=",
  "errorCode": 8101
}
```

PERM_FAIL — Missing mandatory tags:

```
{
  "errorText": "Signature has missing mandatory tag(s): [b, bh]",
  "errorCode": 8101
}
```

TEMP_FAIL — DNS resolution failure:

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
{
  "errorText": "Unable to resolve 809b6e65-a6e7-40f6-8b52-04dd65b6fce1._domainkey.gplb-test.nowyoyo.net. TXT entry does not exist.",
  "errorCode": 8102
}
```

TEMP_FAIL — Cannot extract public key:

```
{
  "errorText": "Unable to resolve 809b6e65-a6e7-40f6-8b52-04dd65b6fce1._domainkey.gplb-test.nowyoyo.net. Cannot get public key",
  "errorCode": 8102
}
```

3.10 Replay Attack Protection

The `X-CSF-SIGNATURE-DATESTAMP` header provides protection against replay attacks. The timestamp is embedded in the encrypted signature and cannot be tampered with independently.

Receiving MAPs SHOULD validate the timestamp based on the message type — rejecting messages with timestamps that are unreasonably old.

3.11 Passthrough Headers

If a MAP is passing a message on to a third party (e.g., TOTSCo operating as a SforB HMAP forwarding to its CPs), the `X-CSF-SIGNATURE` header SHOULD be left intact. This provides additional trust that the message has not been tampered with since it left the originating CP.

3.12 Route Tracking

`X_CSF_ROUTE` is an optional but recommended HTTP header that records the route a message has taken through the network.

Format: `{Epoch milliseconds} {MAP Name}` (alphanumerics only: `[A-Za-z0-9]`)

When a message passes through a MAP, a **new** `X_CSF_ROUTE` header SHOULD be added (not appended to existing headers).

Examples:

```
X_CSF_ROUTE: 1741947045684 TheYellowMap
X_CSF_ROUTE: 1741947056789 AnotherMap
```

In most cases there will be a single entry. When a message passes through a forwarding MAP (e.g., TOTSCo), multiple headers will be present, allowing the final destination to diagnose the route taken while still validating (using PKI) that the message came from the originator.

3.13 Outbound Conflict Resolution

The DNS records used for message signing also resolve routing conflicts. If a CP appears in more than one MAP directory simultaneously (e.g., during a MAP transition):

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

1. The sending MAP looks up `[RCPID]._mapkey.[cp-domain]` in DNS
2. The DNS record indicates the correct MAP endpoint
3. The message is routed to the MAP indicated by DNS

It is the CP's responsibility to update their DNS entries and remove outdated records that could result in invalid routing.

> NOTE
> : CPs having more than one MAP during transition MUST have a new RCPID with the new MAP.

3.14 Testing Tool

A verification and testing tool is available for developers implementing PKI signing:

Documentation: CSF Verification Testing Endpoint

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

4. Message API (Letterbox)

4.1 Overview

The CSF defines a REST API — the **Letterbox API** — for sending and receiving messages between MAPs. This API is compatible with the TOTSCo Hub letterbox interface as defined in the TOTSCo Hub API Specification v2.0.

The Letterbox API is **synchronous** in its HTTP interaction: when a message is sent, the API replies within the same session. However, the messaging model is **asynchronous** — the HTTP 202 response acknowledges receipt, not processing completion.

4.2 API Specification

FIELD	VALUE
API Name	CSF-LetterboxAPI
HTTP Method	POST
URI Pattern	`https://{fqdn}/letterbox/{version}/post`
Transport Security	HTTPS, TLS 1.3
Authentication	OAuth 2.0 Bearer Token
Request Format	`application/json`
Request Headers	`Authorization`, `Accept`, `Content-Type`, `X-CSF-SIGNATURE`, `X-CSF-SIGNATURE-DATESTAMP`

URI Elements

ELEMENT	DESCRIPTION	FORMAT
`fqdn`	Fully Qualified Domain Name of the MAP's letterbox endpoint	RFC 1035 compliant
`version`	API version number	`n.n` (e.g., `1.0`)

4.3 Message Structure

Every message consists of two parts:

1. **Envelope** — Delivery instructions (source, destination, routing, audit data). Defined by the JAM Specification.
2. **Message Body** — The industry process content (e.g., SforB Switch Match). Defined by the relevant process specification.

The CSF is responsible only for the **envelope**. The message body is opaque to the transport layer unless the MAP provides a hosted solution (portal, matching service) on behalf of its CPs.

Envelope Example

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

The following JSON example uses the structure from the JAM Specification. Each `auditData` entry is a `{ "name": "<String>", "value": "<String>" }` pair — both fields are strings and use double-quoted keys consistently. All JSON examples in this document have been audited for that consistency.

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "d8322d80-92c8-4906-a408-f1d2daf7e03d",
      "correlationID": "XYZ987"
    },
    "destination": {
      "type": "RCPID",
      "identity": "7d87c2a7-4f64-433c-8ae1-9460f19a0a13",
      "correlationID": "ABC123"
    },
    "routingID": "businessSwitchMatchRequest",
    "auditData": [
      {
        "name": "auditFieldName",
        "value": "auditFieldValue"
      }
    ]
  },
  "_messageBody": {
    "_comment": "Process-specific message content goes here"
  }
}
```

Envelope Elements

ELEMENT	DESCRIPTION	TYPE	REQUIRED
`envelope`	Container for delivery information	Object	Yes
`source`	Originator of the message and return address	Object	Yes
`destination`	Intended recipient of the message	Object	Yes
`source/type`, `destination/type`	The directory list type (e.g., `RCPID`)	String	Yes
`source/identity`, `destination/identity`	The RCPID of the sender/recipient	String	Yes
`source/correlationID`	Unique identifier for request correlation. MUST always be provided in source.	String	Yes
`destination/correlationID`	Present only in response messages — contains the original sender's correlationID	String	Yes in Response/Failure messages
`routingID`	Determines message type and routing. Must be a value supported by the recipient MAP's routing groups.	String	Yes
`auditData`	Name-value pairs for reporting and auditing	Array	Optional
`auditData[]/name`	The audit field name	String	Yes (within array)
`auditData[]/value`	The audit field value	String	Yes (within array)

> **NOTE**
> : The actual message body eelement name MUST match the `routingID` (e.g., `businessSwitchMatchRequest`). The `_messageBody` key shown above is illustrative.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Audit Data Conventions

Two CSF-wide uses for `auditData` shape how the field SHOULD be populated by senders and read by receivers:

- **Test traffic flagging.** Messages originated by, or routed on behalf of, a `TEST`-status CP or `TEST`-status MAP MUST include `{ "name": "test", "value": "true" }`. The receiving MAP MUST preserve this flag end-to-end and MUST exclude such messages from production telemetry. The full rules — including the MAP-overrides-CP behaviour and the per-message exclusion criteria — are defined in Onboarding & Testing Process §5 — Test Traffic Flagging in the JAM Envelope and §5.1 — Telemetry Exclusion.
- **Match-session sequencing for telemetry.** A SforB Switch Match journey often spans several `businessSwitchMatchRequest` attempts as the GRCP corrects its data. Session and attempt metadata SHOULD be carried in `auditData` so multi-attempt journeys can be aggregated as a single customer journey when MAPs compute industry KPIs, rather than each attempt inflating the failure rate. The recommended `match*` / `serviceMatch*` / `lrcp*` / `map*` field-name prefixes and the three-layer GRCP/LRCP/MAP scheme are summarised in Onboarding & Testing Process §5, with the full proposal in REVIEW.md §6.5 — Match Session Sequencing.

4.4 Successful Response

On successful validation and acceptance, the API returns:

HTTP 202 Accepted

> NOTE
> "The request has been accepted for processing, but the processing has not been completed. The request might or might not be eventually acted upon and may be disallowed when processing occurs."

No JSON body is returned on a successful 202 response.

Validation Performed Before 202

The receiving MAP validates the following:

1. OAuth 2.0 token validity
2. JSON message structure
3. Envelope attribute validity
4. Source type and RCPID validity
5. Source RCPID account status
6. Destination type and RCPID validity
7. Destination RCPID account status
8. **Source MAP authorisation via `_mapkey` DNS verification** — see §4.4.1 below.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

9. Routing ID validity
10. DKIM signature verification (via `X-CSF-SIGNATURE`)

The MAP could perform these in any order, but the MAP MUST NOT respond until all checks have passed.

4.4.1 Per-Message Source-MAP Verification (Step 8 detailed)

For every inbound message, in addition to the DKIM signature verification defined in §3.4, the receiving MAP MUST verify that the originating MAP is the one currently authorised by the source CP in DNS:

1. The receiving MAP MUST resolve `[sourceRCPID]._mapkey.[source-cp-domain]` in DNS, where `sourceRCPID` is taken from `envelope.source.identity` and `source-cp-domain` is the signing domain from the DKIM `d=` tag in the `X-CSF-SIGNATURE` header.
2. The receiving MAP MUST extract the originating MAP's letterbox URL — for direct CSF-MAP-to-CSF-MAP traffic this is the MAP that established the TLS connection and presented the OAuth bearer token; for HMAP-forwarded traffic, it is the first non-HMAP entry in the `X_CSF_ROUTE` header chain (see §4.4.2 below).
3. If the resolved `_mapkey` URL does not match the originating MAP's letterbox URL (compared on host + path, **case-insensitive on host, case-sensitive on path**, trailing-slash-insensitive), the receiving MAP MUST reject the message with HTTP 403 and CSF error code **`9301`** ("DNS `_mapkey` does not resolve to the originating MAP"). The asynchronous delivery-failure record MUST include the resolved `_mapkey` value (for diagnostics).
4. DNS resolutions performed for `_domainkey` (§3.4) and `_mapkey` (this section) MUST honour the published TTL and SHOULD be served from the MAP's local DNS server per §3.3 — MAP Responsibilities (RFC 6303).

> NOTE ON `9301` SCOPE.
> Error code `9301` was previously documented as the reject for `rcpidStatusRequest` authorisation failures only (§7.2.3). As of CSF v2.x it covers **all** source-MAP authorisation failures, including this per-message verification. Existing §7.2.3 usage is unchanged and is a specialisation of this general rule.

> NOTE ON TOTSCO RESPONSE POLICY.
> The `9301` code is a CSF-specific failure. When operating as a Hub MAP (HMAP), TOTSCo does not implement DKIM signing (§3.6) and does not perform this check; CSF MAPs MUST NOT include `9301` in responses sent back to TOTSCo Hub traffic (§4.6). This is consistent with how CSF 8xxx codes are excluded from TOTSCo responses.

4.4.2 HMAP-Forwarded Traffic

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

When a message arrives from an HMAP (e.g. TOTSCo), the originating MAP for the purposes of §4.4.1 is the **first non-HMAP entry in the `X_CSF_ROUTE` header chain**, not the HMAP itself. HMAPs do not implement DKIM signing (§3.6) and do not appear in any CP's `_mapkey` record — the `_mapkey` check therefore does not apply to the HMAP. It applies only to the underlying CSF MAP that originated the message into the HMAP.

For HMAP-only traffic where no CSF MAP appears in the route (i.e. a message originated entirely within a hub-and-spoke network), §4.4.1 is not enforced; the integrity guarantee for such traffic rests on the HMAP's own commercial controls and existing TOTSCo authentication. The boundary between HMAP-only and CSF-originated traffic is determined by the presence or absence of CSF MAP entries in `X_CSF_ROUTE`.

> TRUST ASSUMPTION FOR `X_CSF_ROUTE`.
> The chain is treated as trustworthy on the basis of (a) the OAuth-authenticated channel each hop uses and (b) the bilateral trust model between MAPs that have completed onboarding. Signing of `X_CSF_ROUTE` entries by each MAP that appends to the chain is a future enhancement and not required for this clause.

4.5 Synchronous Error Responses

HTTP Error Codes

HTTP CODE	ERROR	DESCRIPTION
400	Bad Request	Invalid JSON, missing required fields, or validation error
401	Unauthorised	Authentication failure (invalid OAuth token or API key)
403	Forbidden	Valid credentials but insufficient authorisation, or PKI verification failure
404	Not Found	Requested resource not found
405	Method Not Allowed	HTTP method not supported (e.g., GET instead of POST)
429	Too Many Requests	Rate limit exceeded
500	Internal Server Error	Unexpected processing error
501	Not Implemented	HTTP method not supported by the server
502	Bad Gateway	Invalid response from upstream
503	Service Unavailable	Temporary maintenance
504	Gateway Timeout	Upstream response timeout

Validation Error Codes (9xxx Series)

These codes are returned in the JSON error response body alongside the appropriate HTTP status code:

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

STEP	VALIDATION	HTTP CODE	ERROR CODE
1	OAuth 2.0 token validation	401	N/A
2	JSON message structure	400	N/A
3	Envelope attribute validation	400	N/A
4	Source type validity	400	9002
5	Source RCPID validity	400	9003
6	Source RCPID account status	403	9003
7	Destination type validity	400	9000
8	Destination RCPID validity	400	9001
9	Destination RCPID account status	403	9001
10	Source MAP authorisation via `_mapkey` DNS check (§4.4.1)	403	9301
11	Routing ID mapping to source RCP	400	9010
12	Routing ID validity	400	9012

PKI Error Codes (8xxx Series)

ERROR CODE	MEANING	RETRY?
8101	PERM_FAIL — Signature verification failed permanently	No
8102	TEMP_FAIL — Signature verification failed due to temporary issue (e.g., DNS)	Yes

Synchronous Error Codes (9xxx Series)

ERROR CODE	MEANING
9000	Unknown or invalid destination Type.
9001	Unknown or invalid destination ID.
9002	Unknown or invalid source Type.
9003	Unknown or invalid source ID.
9301	DNS `_mapkey` does not resolve to the originating MAP — source-MAP authorisation failed (§4.4.1). Also used for `rcpidStatusRequest` authorisation failures (§7.2.3). Not sent to TOTSCo Hub traffic.

Error Response Format

For validation errors:

```
{
  "errorCode": "9002",
  "errorText": "Unknown or invalid source Type."
}
```

For HTTP-level errors:

```
{
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
  "code": "400",
  "message": "Bad Request",
  "description": "Schema validation failed in the Request: [Path '/directory'] Object has missing required properties
([\"listID\"])"
}
```

The original documentation uses two different error response formats — one with `errorCode`/`errorText` for validation errors (9xxx,8xxx) and another with `code`/`message`/`description` for HTTP errors. Implementers should support both formats.

4.6 Asynchronous Error Handling — Message Delivery Failures

When a MAP is unable to deliver a message to its intended recipient, it creates a failure notification message back to the originator using the following format:

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "MAP-X-IDENTITY"
    },
    "destination": {
      "type": "RCPID",
      "identity": "ORIGINAL-SOURCE-RCPID",
      "correlationID": "ORIGINAL-CORRELATION-ID"
    },
    "routingID": "messageDeliveryFailure",
    "auditData": [
      { "name": "originalDestinationType", "value": "RCPID" },
      { "name": "originalDestination", "value": "DESTINATION-RCPID" },
      { "name": "originalRoutingID", "value": "businessSwitchMatchRequest" },
      { "name": "faultCode", "value": "9008" }
    ]
  },
  "messageDeliveryFailure": {
    "code": "9008",
    "text": "Unable to deliver the message to the destination, timed out.",
    "severity": "failure"
  }
}
```

Delivery Failure Fields

FIELD	DESCRIPTION	TYPE
`messageDeliveryFailure`	Container for delivery failure details	Object
`code`	Numeric fault code for remedial action	Integer
`text`	Human-readable description of the fault	String
`severity`	Nature of the notification: `information`, `warning`, or `failure`	String

> NOTE
> : The source does not contain a `correlationID` because delivery failure notifications cannot be replied to. The originator's correlation ID is returned in the destination field.

Delivery Failure Codes (9xxx Series)

CODE	DESCRIPTION	SEVERITY
9005	Unable to deliver — no valid route	Failure

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

CODE	DESCRIPTION	SEVERITY
9006	Unable to deliver — rejected, invalid message format (HTTP 400 from destination)	Failure
9007	Recipient rejected message (HTTP 404 from destination)	Failure
9008	Unable to deliver — timed out (HTTP 5xx or timeout after retries)	Failure
9013	Unable to deliver — invalid API key	Failure
9014	Unable to deliver — API key expired	Failure
9015	Unable to deliver — digital certificate invalid	Failure
9016	Unable to deliver — digital certificate expired	Failure

The CSF-specific 8xxx series is complete with `8101` (PERM_FAIL) and `8102` (TEMP_FAIL) — no further CSF 8xxx delivery failure codes are required (agreed by the TAG). When responding to TOTSCo, CSF-specific 8xxx codes MUST NOT be included in the response, since TOTSCo does not recognise them and only the industry-standard 9xxx series applies on that interface.

ERROR CODE	MEANING	RETRY?
8101	PERM_FAIL — Signature verification failed permanently	No
8102	TEMP_FAIL — Signature verification failed due to temporary issue (e.g., DNS)	Yes

4.7 Message Delivery Policy

This section ratifies the CSF's message delivery policy for MAP-to-MAP exchanges. It also marks the boundary at which TOTSCo's hub-side principles (the OTA2 "GPLB Switching for Business Message Delivery Principles") bind on the TOTSCo side — see §4.7.4.

The policy was formalised by the TAG following the comparative analysis against the OTA2 principles document (see the CSF-vs-GPLB-MDP Analysis Report). The architectural framing in §4.7.1 below explains why CSF retains its own P2P delivery policy rather than adopting the GPLB hub schedules wholesale.

4.7.1 Architectural framing — synchronous P2P vs hub boundary

The CSF is a **peer-to-peer synchronous** transport. Two MAPs exchange JAM messages directly over TLS 1.3 / OAuth 2.0 / DKIM-signed HTTP. The sending MAP knows the result of its delivery attempt **in the same HTTP exchange**:

- `HTTP 202` confirms the receiving MAP has accepted the message for downstream processing.
- `HTTP 4xx` plus a synchronous `9xxx` industry error code confirms the receiving MAP has rejected it for a documented reason.
- A network / connection / response timeout signals that the receiving MAP is unreachable.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

This is materially different from a centralised hub model — where the hub sits between sender and destination and absorbs the delivery asynchrony, queueing, retrying on a per-protocol schedule, ageing stale queue items, and emitting asynchronous failure notifications back to the sender on a separate channel. **CSF MAPs do none of that at the framework layer.** The synchronous response IS the delivery indication for the transport step.

The asymmetry has one important boundary: the **HMAP interface** between a CSF MAP and TOTSCo. TOTSCo is a hub by its own design, so traffic crossing that boundary follows the GPLB Message Delivery Principles on TOTSCo's side. See §4.7.4 below.

4.7.2 Required behaviours (CSF MAP-to-MAP)

When a sending MAP attempts to deliver a JAM message to a receiving MAP's letterbox, the following are required:

- **Connection timeout: 1 second.** TCP / TLS handshake to the destination MUST succeed within 1 second or the attempt is treated as a failed delivery and feeds into the retry loop.
- **Response timeout: 3 seconds.** Once connected, the sender waits 3 seconds for `HTTP 202`. If no response is received within 3 seconds, the attempt is treated as a failed delivery and feeds into the retry loop.
- **Retry with exponential backoff and jitter.** See §4.7.5 for the architectural rationale. MAPs MUST NOT use linear or fixed-cadence retry schedules at the CSF P2P boundary.
- **Respect HTTP semantics.** MAPs MUST honour `503 Service Unavailable` and `429 Too Many Requests` with the destination's `Retry-After` header. A `Retry-After` value overrides the sender's normal backoff schedule for that attempt.
- **Circuit-breaker pattern.** MAPs MUST implement a circuit-breaker: after **N consecutive failures** to the same destination (recommended `N = 5`, tunable per MAP), the sender SHOULD stop active retry, fail subsequent messages fast for a recovery period, and raise an operational alert.
- **Bounded maximum retry period.** MAPs MUST NOT "wait forever". The framework recommends a maximum retry window in the order of **hours, not days** — typically 1–4 hours for production traffic. Persistent failure beyond the window triggers the escalation in §4.7.3.
- **All delivery attempts MUST be logged.** Log entries include the destination URL, timestamp, attempt number, outcome, and any synchronous error code returned. Audit retention follows §9 audit-log requirements.

4.7.3 Operational-engagement escalation

When a destination MAP fails persistently beyond the sender's retry window, the CSF's primary recovery mechanism is **operational engagement between MAP operations teams** — not silent retry. The procedure:

1. The failing MAP publishes a `map/outage[]` entry in its CP Registry (per §5.4 Field Reference). Every other MAP picks it up on the next directory poll and applies the appropriate downgrade to retry behaviour (see Part 2 §9.3 Outage Notification).

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

2. The sender raises an internal alert when its retry window is exhausted. The alert records the destination RCPID, the duration of consecutive failure, and the last synchronous error code observed.
3. Operations teams from both sides engage via the published `map/contact[]` channels (phone, email, ticketing URL — see §5.6 CP Contact Object). Contact entries include availability hours, days, and purpose tags so the sender knows which channel is appropriate at the time of the incident.
4. The framework's standard incident response process applies — see Part 1 §10.6.5 Incident Response Coordination.

This sequence is the deliberate replacement for "wait 12 days" hub-side queue ageing. **CSF MAPs do not silently retry for days; they escalate.** A persistent destination failure is a P1 / P2 incident, not a routine retry condition.

4.7.4 The HMAP boundary

When a CSF MAP exchanges traffic with TOTSCo via the dedicated HMAP interface, the boundary inherits hub behaviour by definition. TOTSCo is a centralised hub, and the **GPLB Switching for Business Message Delivery Principles V0.2** (OTA2) bind TOTSCo's hub-side delivery to the destination RCP. Specifically:

- The 1 s connection / 3 s response timeouts in §4.7.2 are aligned with the GPLB principles — no asymmetry on those two values.
- TOTSCo applies its own hub-side retry schedules per the GPLB principles — **linear** 5 / 10 / 15 / 20 / 25 s for Match Requests within a **30 s** window; 10 / 20 / 30 / 60 s then a 60 s fixed-cadence tail for other SforB messages within a **12-day** window.
- TOTSCo emits asynchronous failure notifications back to the sender's pre-configured failure endpoint when hub-side delivery fails. The CSF MAP receives those notifications as a standard hub client.
- TOTSCo operates the single-flight-per-endpoint queue and queue ageing on its hub side.

The HMAP interface is locked down per [§6.3.1](06-totsco-integration.md#631-dedicated-locked-down-hmap-interfaces--both-directions). CSF version increments — including any future change to §4.7 itself — do **NOT** propagate through the dedicated TOTSCo endpoints. The asymmetric behaviour at the HMAP boundary is stable and bilateral by design, so the GPLB hub schedules do not bleed back into the wider CSF P2P transport.

For the full operating model — including the per-MAP dedicated endpoint pattern, the worked NowYoYo / Red Company bidirectional example, and the locked-down interface architectural commitment — see the standalone TOTSCo HMAP Integration Guide.

4.7.5 Why exponential backoff (and not linear / fixed-cadence retry)

The CSF retains exponential backoff with jitter as its P2P default. The linear and fixed-cadence retry schedules in the OTA2 GPLB Message Delivery Principles (5 / 10 / 15 / 20 / 25 s for Match Requests; 10 / 20 / 30 / 60 s then a 60 s cadence for other messages) are

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

architecturally appropriate for a single centralised hub but **inappropriate** for a distributed P2P network for two distinct reasons:

1. **Linear schedules synchronise retry storms.** When a destination is recovering, many CSF MAPs hitting it at exactly the same intervals creates the "thundering herd" effect that exponential-with-jitter is specifically designed to prevent. Under linear retry pressure, the destination has a synchronised load spike every 5 s (Match) or every 60 s (other) until it falls back over. Under exponential-with-jitter pressure, the load is spread across the recovery window and the destination is far more likely to recover cleanly.
2. **Exponential-with-jitter is the industry-standard distributed-systems pattern.** Every modern cloud SDK (AWS, Google, Microsoft, Cloudflare, etc.) uses this pattern as its default retry policy. The pattern is documented in the SRE literature and is the recommended approach for any service operating across multiple independent clients. The CSF aligns with that practice; the alignment is deliberate, not incidental.

The framework's principle is therefore: **exponential backoff with jitter, bounded by a maximum retry window, with operational escalation taking over beyond that window.** Specific exponential parameters (base, cap, jitter range) are MAP implementation choices; the SHAPE of the curve — geometric growth with randomisation — is required.

What exponential backoff with jitter means in practice

- **Exponential backoff**: when a request fails, wait a short period, then exponentially grow the delay between subsequent retries (e.g. 1 s, 2 s, 4 s, 8 s, 16 s, capped at some maximum). This gives the struggling destination time to recover instead of repeatedly hammering it.
- **Jitter**: a random component (typically ±25 %) added to each delay, so independent clients spread their retries across the recovery window rather than aligning on identical intervals.

This is the only retry shape that scales gracefully as the CSF network grows. Linear retries do not.

4.7.6 Summary of bindings

RULE	CSF MAP-TO-MAP (P2P)	HMAP BOUNDARY (CSF MAP ⇔ TOTSCO)
1 s connection timeout	MUST	MUST (aligned with GPLB)
3 s response timeout	MUST	MUST (aligned with GPLB)
Exponential backoff with jitter	MUST	TOTSCo applies GPLB linear / fixed-cadence schedule on its hub side; CSF MAP applies exponential on its own retries to TOTSCo
Respect `Retry-After` on 429 / 503	MUST	MUST
Circuit-breaker	MUST	MUST (CSF MAP side); n/a on hub side

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

RULE	CSF MAP-TO-MAP (P2P)	HMAP BOUNDARY (CSF MAP ↔ TOTSCO)
Maximum retry period	MUST be bounded (hours)	TOTSCo applies the GPLB 30 s Match / 12-day other-message holds
Failure notification to sender	Synchronous — `9xxx` / `8xxx` in the original response	Asynchronous from TOTSCo when hub-side delivery fails; CSF MAP consumes as a standard hub client
Operational escalation via `map/contact[]` / `map/outage[]`	Primary recovery mechanism beyond the retry window	Applies on the CSF MAP side; TOTSCo follows its own incident process
Logging of every delivery attempt	MUST	MUST

For test cases that exercise these behaviours, see Onboarding & Testing Process §8.7 Retry & Delivery Policy.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

5. CP Registry API

5.1 Overview

Each MAP publishes a **CP Registry** — a JSON document containing everything another MAP needs to know about the publishing MAP and its connected CPs. Other MAPs collect these CP Registries via a REST API endpoint and converge them into a private **Master Registry**. From the Master Registry, each MAP derives a filtered **Directory** for its own CPs.

> DESIGN INTENT — SUPERSET OF TOTSCO'S DIRECTORY.
> The CSF CP Registry is **deliberately a strict superset of the TOTSCo OTS/GPLB directory structure** so that TOTSCo (and any party that already consumes the TOTSCo directory) can ingest the CSF Registry with its existing parser. CSF-specific additions — `signing[]`, expanded `resource[]`, `contact[]`, `outage[]`, and the routing-group metadata in §5.4.1 — are additive. Parsers that don't recognise these fields ignore them; the underlying TOTSCo-compatible shape is unchanged. This was an explicit design decision intended to make industry-wide consumption trivial through the JSON contract.

5.1.1 Side-by-side — TOTSCo vs CSF registry shape

TOTSCo format (OTS / GPLB today):

```
"list": [
    {
        "listType": "RCPID",
        "identity": [
            {
                "id": "RTLL",
                "name": "NowYoYo",
                "processSupport": [
                    {
                        "process": "OTS",
                        "status": "ACTIVE"
                    }
                ],
                "resource": [
                    {
                        "name": "salesAssistURL",
                        "type": "URL",
                        "value": "https://nowyoyo.com/ots/sales-assistance/"
                    },
                    {
                        "name": "customerAssistURL",
                        "type": "URL",
                        "value": "https://nowyoyo.com/ots/customer-assistance/"
                    }
                ]
            }
        ]
    }
]
```

CSF format (superset — extra fields are additive, never breaking):

```
"list": [
    {
        "identity": [
            {
                "id": "c42569a5-d0ed-46f7-b9df-4ea0ba6a29c8",
                "name": "Red Company ISP",
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
            "processSupport": [
                {
                    "process": "GPLB",
                    "status": "ACTIVE"
                }
            ],
            "resource": [
                {
                    "name": "customerAssistURL",
                    "type": "URL",
                    "value": "https://gplb.red-company.com/customerAssistURL.html"
                }
            ],
            "signing": [
                {
                    "domain": "gplb-test.nowyoyo.net",
                    "routingID": [
                        "businessSwitch.*"
                    ]
                }
            ]
        }
    ]
  }
]
```

> **NOTE**
> The `id` is wider in the CSF (UUIDv4 instead of 4-character) but both are valid `string` values under the JAM Specification (up to 256 characters). The `signing[]` block is CSF-only; TOTSCo can ignore it. CSF MAPs publish the same `processSupport[].process` values (`OTS`, `GPLB`) that TOTSCo already uses, so process-based filtering is identical on both sides.

5.1.2 Dual-process filtering — same URL pattern across both networks

TOTSCo's pre-production Directory v2 already combines OTS and GPLB entries in one list, distinguishing them by `processSupport[].process`. MAPs that want a single-process view filter on an `identity` query parameter:

```
https://preprod.otshub.totsco.co.uk/directory/v2/entry?listType=RCPID&identity=all
https://preprod.otshub.totsco.co.uk/directory/v2/entry?listType=RCPID&identity=ots
https://preprod.otshub.totsco.co.uk/directory/v2/entry?listType=RCPID&identity=gplb
```

We assume the same filter pattern in production. The CSF's Master Registry follows the same convention — `processSupport[].process` carries `OTS`, `GPLB`, and future process tags — so any TOTSCo-side filtering logic carries over unchanged when consuming a CSF CP Registry.

5.2 API Specification

FIELD	VALUE
HTTP Method	GET
Authentication	OAuth 2.0 Bearer Token (same credentials as messaging)
Response Format	`application/json`
Transport Security	HTTPS, TLS 1.3

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

5.3 Registry JSON Structure

The registry has two main sections: `map` (MAP metadata) and `list` (CP directory entries).

Complete Example

```
{
  "map": {
    "name": "myMap",
    "version": "2024-01-01T13:45:24Z00000",
    "registrationURI": "https://apply.mymapdomain.com/register",
    "serviceStatusURI": "https://status.mymapdomain.com/myStatus",
    "serviceStatusServiceURI": "https://status.mymapdomain.com/myStatusAPI",
    "contact": [
      {
        "type": "phone",
        "value": "01111 111111",
        "purpose": ["support", "sales"],
        "availabilityFrom": "09:00",
        "availabilityTo": "17:00",
        "availabilityDays": ["monday", "tuesday", "wednesday", "thursday", "friday"],
        "comment": "Use for all support calls. Support hours 9am to 5pm."
      },
      {
        "type": "email",
        "value": "support@myMapDomain.com"
      },
      {
        "type": "url",
        "value": "https://support.mymapdomain.com/support"
      }
    ],
    "connectivity": {
      "directory": {
        "url": "https://api.mymapdomain.com/directory",
        "oauthTokenURL": "https://api.mymapdomain.com/token"
      },
      "services": [
        {
          "description": "Endpoint for OTS messages",
          "service": "messaging",
          "url": "https://api.mymapdomain.com/letterboxOTS",
          "oauthTokenURL": "https://api.mymapdomain.com/token",
          "routingGroups": ["OTSMessages"]
        },
        {
          "description": "Endpoint for SforB (GPLB) messages",
          "service": "messaging",
          "url": "https://api.mymapdomain.com/letterboxGPLB",
          "oauthTokenURL": "https://api.mymapdomain.com/token",
          "routingGroups": ["GPLBMessages"]
        }
      ]
    },
    "routingGroups": [
      {
        "name": "OTSMessages",
        "routingID": ["residentialSwitch.*"]
      },
      {
        "name": "GPLBMessages",
        "routingID": ["businessSwitch.*"]
      }
    ],
    "connections": [
      {
        "name": "MAP1",
        "registrationURI": "https://apply.map1.com/register"
      },
      {
        "name": "MAP2",
        "registrationURI": "https://apply.map2.com/register"
      }
    ],
    "sponsorship": {
      "order": 3,
      "nextSponsor": 4,
      "nextSponsorUpdated": "2026-06-09T10:00:00Z",
      "capacity": 2,
      "registrationURI": "https://apply.mymapdomain.com/become-a-map",
      "active": [
        {
          "id": "a3f91c2d-7e4b-4f8a-b1c2-9d3e5f6a7b8c",
          "type": "traditional",
          "opened": "2026-06-09T10:00:00Z",
          "candidateRef": "ONB-2026-0042"
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
        }
      ]
    }
  },
  "list": [
    {
      "listType": "RCPID",
      "identity": [
        {
          "id": "d8322d80-92c8-4906-a408-f1d2daf7e03d",
          "name": "Xenon Telecom",
          "processSupport": [
            { "process": "OTS_Porting", "status": "Active" },
            { "process": "GPLB", "status": "Active" }
          ],
          "resource": [
            { "name": "customerAssistURL", "type": "URL", "value": "https://xenon.com/SforB" },
            { "name": "salesAssistURL", "type": "URL", "value": "https://xenon.com/sales" }
          ],
          "signing": [
            {
              "domain": "ots.xenon.com",
              "routingID": ["residentialSwitch.*"]
            },
            {
              "domain": "gplb.xenon.com",
              "routingID": ["businessSwitch.*", "notification"]
            }
          ],
          "contact": [
            {
              "type": "phone",
              "value": "020 7946 0123",
              "purpose": ["sales"],
              "availabilityFrom": "09:00",
              "availabilityTo": "17:00",
              "availabilityDays": ["monday", "tuesday", "wednesday", "thursday", "friday"],
              "comment": "Sales team available weekdays 9am-5pm"
            },
            {
              "type": "email",
              "value": "switching-support@xenon.com",
              "purpose": ["support", "technical"],
              "comment": "24/7 email support — response within core business hours"
            },
            {
              "type": "url",
              "value": "https://xenon.com/switching-support",
              "purpose": ["support"],
              "comment": "Online ticketing system for switch query resolution"
            }
          ]
        }
      ]
    }
  ]
}
```


TOTSCo maps the CSF Source UUIDv4 values to their own pseudo-created 4-digit character equivalents, which can be done transparently with its own managed clients. The CSF network does not need to know what these 4-digit characters are. When TOTSCo sends a message to the CSF, they will translate their own Source 4-digit RCPID to the required CSF UUIDv4 RCPID before routing the message using the Master CSF directory that they have compiled.

MAP in TEST Mode — Onboarding Example

When a new MAP is undergoing onboarding, its registry SHOULD include `map.status` set to `TEST`. Receiving MAPs MUST treat every CP under that MAP as test, regardless of the per-CP `processSupport[].status`, and exclude those CPs from the CP-facing Directory until the MAP transitions to `ACTIVE`. See Onboarding & Testing Process §4.2.


```
{
  "map": {
    "name": "newMap",
    "status": "TEST",
```


© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
    "version": "2026-04-29T10:30:00Z00000",
    "registrationURI": "https://apply.newmapdomain.com/register",
    "...": "remainder of map object as per the main example"
  },
  "list": [ "..." ]
}
```

> **NOTE**
> : `map.status` is optional. If absent, the MAP is treated as `ACTIVE`. The valid values are `ACTIVE`, `TEST`, and `SUSPEND`.

5.4 MAP Section — Field Reference

JSON PATH	DESCRIPTION	TYPE	REQUIRED
`map`	Container for MAP metadata	Object	Yes
`map/name`	MAP name	String	Yes
`map/status`	MAP-level operational status: `ACTIVE`, `TEST`, `SUSPEND`. If absent, treated as `ACTIVE`. When set to `TEST`, all CPs under this MAP are treated as test regardless of their own per-CP `processSupport[].status`. See Onboarding & Testing Process §4.2.	String (enum)	Optional
`map/version`	Last updated timestamp. Used to determine if local cached values need changing	DateTime (`yyyy-MM-ddTHH:mm:ssZSSSSS`)	Yes
`map/registrationURI`	URI for CP/MAP registration	String	Optional
`map/serviceStatusURI`	URI for operational status page	String	Optional
`map/serviceStatusServiceURI`	API URI for programmatic status checks	String	Optional
`map/contact[]`	Contact methods for the MAP	Object Array	Yes
`map/contact[]/type`	Contact type: `phone`, `email`, `url`	String (enum)	Yes
`map/contact[]/value`	Contact value (number, address, or URL)	String	Yes
`map/contact[]/purpose[]`	What the contact is for: `support`, `sales`, `technical`	String Array	Optional
`map/contact[]/availabilityFrom`	Availability start time	Time (`HH:mm`)	Optional
`map/contact[]/availabilityTo`	Availability end time	Time (`HH:mm`)	Optional
`map/contact[]/availabilityDays[]`	Days available	String Array (day names)	Optional
`map/contact[]/comment`	Usage notes	String	Optional
`map/outage[]`	Array of current outage or planned maintenance entries. An empty array or absent field signals GREEN — no reported issues. MAPs implementing outage	Object Array	Yes (empty if no issues)

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

JSON PATH	DESCRIPTION	TYPE	REQUIRED
	notification SHOULD always include this field, even as an empty array. See §9.4 Outage Notification.		
`map/outage[]/priority`	ITIL incident priority: `P1` (MAP unavailable — fix within 4 working hours), `P2` (CP unavailable — fix within 1 working day), `P3` (service degradation — fix within 2 working days). Primary severity indicator; replaces traffic light model. See §9.4.2.	String (enum)	Yes
`map/outage[]/type`	Outage type: `PLANNED` or `INCIDENT`	String (enum)	Yes
`map/outage[]/start`	Start time of the outage or maintenance window (ISO 8601 UTC)	DateTime	Yes
`map/outage[]/end`	Expected or actual end time (ISO 8601 UTC). Omit for active incidents where resolution time is unknown.	DateTime	Optional
`map/outage[]/affectedServices`	Routing group names (matching `map/routingGroups[]/name`) affected by this entry. If absent or empty, all services are considered affected.	String Array	Optional
`map/outage[]/description`	Human-readable status narrative: what is affected, current status, ETA (if known), and any workaround. Updated as the incident evolves.	String	Yes
`map/outage[]/contact`	Escalation contact specific to this outage. Uses the same structure as a `map/contact[]` entry.	Object	Optional
`map/outage[]/contact/type`	Contact type: `phone`, `email`, or `url`	String (enum)	Yes (if contact present)
`map/outage[]/contact/value`	Contact value (number, address, or URL)	String	Yes (if contact present)
`map/outage[]/contact/comment`	Usage instructions — e.g. reference numbers, bridge PINs, or when to use this contact	String	Optional
`map/connectivity`	Connection details	Object	Yes
`map/connectivity/directory/url`	Directory API endpoint	String	Yes
`map/connectivity/directory/oauthTokenURL`	OAuth token endpoint for directory access	String	Yes
`map/connectivity/services[]`	Service endpoints	Object Array	Yes
`map/connectivity/services[]/description`	Human-readable description	String	Yes
`map/connectivity/services[]/service`	Service type (e.g., `messaging`)	String	Yes
`map/connectivity/services[]/url`	Service endpoint URL	String	Yes
`map/connectivity/services[]/oauthTokenURL`	OAuth token endpoint for this service	String	Yes

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

JSON PATH	DESCRIPTION	TYPE	REQUIRED
`map/connectivity/services[]/routingGroups[]`	Routing groups served by this endpoint	String Array	Yes
`map/connectivity/services[]/relay`	Whether this endpoint supports message relay	Boolean	Yes
`map/routingGroups[]`	Routing group definitions	Object Array	Yes
`map/routingGroups[]/name`	Routing group name (matches service routing groups)	String	Yes
`map/routingGroups[]/routingID[]`	Accepted routing ID patterns (regex)	String Array	Yes
`map/connections[]`	Other MAPs this MAP has connectivity with	Object Array	Yes
`map/connections[]/name`	Name of the connected MAP	String	Yes
`map/connections[]/registrationURI`	Registration URI of the connected MAP	String	Yes
`map/sponsorship`	Sponsor rotation metadata for the decentralised onboarding of new MAPs. See §5.4.2 for the rotation rules and the Onboarding & Testing Process §3.4 for the full mechanism.	Object	Yes
`map/sponsorship/order`	This MAP's permanent FIFO position in the rotation. Assigned at onboarding; **never changes** — preserved for historical audit consistency even when the MAP withdraws or its `map/status` becomes non-`ACTIVE`. The rotation skips entries whose `map/status` is not `ACTIVE`.	Integer	Yes
`map/sponsorship/nextSponsor`	The `order` number of the MAP designated as the **next** Sponsor after this MAP. Updated when this MAP accepts a new sponsorship engagement.	Integer	Yes
`map/sponsorship/nextSponsorUpdated`	Timestamp when `nextSponsor` was last updated (ISO 8601 UTC). The MAP with the most recent `nextSponsorUpdated` is the current active Sponsor.	DateTime	Yes
`map/sponsorship/capacity`	Maximum concurrent sponsorships this MAP will accept. **Minimum: `1`; no upper bound** — set to reflect the MAP's own operational throughput. A MAP that primarily takes `single-cp` (MAP-of-1) engagements MAY support a higher `capacity` because those complete faster than traditional multi-CP onboardings.	Integer	Yes
`map/sponsorship/registrationURI`	URI advertising the Sponsor MAP's onboarding application page for prospective MAPs (self-service portal, contracted form, or structured registration endpoint). Omitted if the MAP uses non-electronic registration (email,	URI	Optional

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

JSON PATH	DESCRIPTION	TYPE	REQUIRED
	telephone, ticketing); prospective MAPs then use the MAP's `map/contact[]` entries as the contact channel.		
`map/sponsorship/active[]`	Array of currently open sponsorship engagements. Empty array when idle.	Object Array	Yes
`map/sponsorship/active[]/id`	UUIDv4 identifier for this engagement, generated by the Sponsor MAP on acceptance.	String (UUID)	Yes
`map/sponsorship/active[]/type`	Type of candidate: `traditional` (multi-CP MAP) or `single-cp` (MAP onboarding with a single CP, e.g. a MAP-of-1 migrating from TOTSCo SforB per Part 1 §7.4). Both types count as **`1`** against the MAP's `capacity`.	String (enum)	Yes
`map/sponsorship/active[]/opened`	Timestamp when the Sponsor MAP accepted the engagement (ISO 8601 UTC).	DateTime	Yes
`map/sponsorship/active[]/candidateRef`	Non-identifying reference assigned by the Sponsor MAP for its own coordination — for example a ticket number from the Sponsor's internal onboarding system when handling several concurrent engagements. Free-form string; not standardised across MAPs; consumers outside the issuing MAP MUST NOT rely on its semantics.	String	Optional

5.4.1 Routing Groups

In the MAP section the `routingGroups` in conjunction with `connectivity/services[]` are used to determine where to send a message. In this snippet of the MAP object, there are 3 routing groups:

```
[
  {
    "name": "OTSMessages",
    "routingID": [
      "resi.*"
    ]
  },
  {
    "name": "GPLBMessages",
    "routingID": [
      "businessSwitch.*"
    ]
  },
  {
    "name": "FutureMessages",
    "routingID": [
      "futureThing.*"
    ]
  }
]
```

The `routingID` is a standard regex pattern that is matched against the `routingID` in the envelope of the message to be sent. The sender uses this to determine the `name` of the

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

correct routing group. So a message with a routing ID of `futureThingActivateRequest` will match with the `FutureMessages` routing group.

This routing group `name` is then matched against the array `connectivity/services[]`, specifically the `routingGroups[]` array.

Below is a snippet of the `connectivity/services[]` MAP object:

```
[
  {
    "description": "Endpoint for OTS messages",
    "service": "messaging",
    "url": "https://api.mymapdomain.com/letterboxOTS",
    "oauthTokenURL": "https://api.mymapdomain.com/token",
    "routingGroups": [
      "OTSMessages"
    ]
  },
  {
    "description": "Endpoint for Other messages",
    "service": "messaging",
    "url": "https://api.mymapdomain.com/letterbox",
    "oauthTokenURL": "https://api.mymapdomain.com/token",
    "routingGroups": [
      "GPLBMessages", "FutureMessages"
    ]
  }
]
```

By using the `name` calculated in the previous step, `FutureMessages` is matched against and resolves to the `Endpoint for Other messages`. This contains a Auth and Message URL allowing the message to sent to the MAP in question.

This is a simple example of how routing groups can be used to determine where to send a message. Some MAPs may use the same endpoints for all messages, while others may a different endpoint for each message type. It provides maximum flexibility.

5.4.2 Sponsorship Object — Rotation Rules

The `sponsorship` object enables MAPs to coordinate a fair, decentralised rotation for onboarding new MAPs onto the CSF. The detailed mechanism (acceptance-advance, chain-walk overflow, edge-case handling, worked examples) is set out in the Onboarding & Testing Process §3.4. This section sets out the rules that bind every MAP's implementation of the `sponsorship` object.

RULE	STATEMENT
Capacity bounds	The minimum permitted `capacity` is **1** — every onboarded MAP MUST be willing to take at least one engagement at a time. There is **no upper bound** on `capacity`; a MAP sets the value to reflect its own operational throughput. A MAP that primarily takes `single-cp` (MAP-of-1) engagements MAY support a higher `capacity` because those engagements complete faster than traditional multi-CP onboardings.
Engagement weighting	A `traditional` engagement and a `single-cp` engagement each count as **`1`** against the MAP's `capacity`. No reduced weighting is applied to `single-cp`. The equal-weighting rule protects small MAPs operating at `capacity: 1`: such a MAP is not expected to take additional engagements simply because some are MAP-of-1s.
`order` persistence	A MAP's `order` is permanent. When a MAP withdraws or its `map/status` becomes non-`ACTIVE`, its `order` is **retained for historical audit consistency** and is not reassigned. The rotation skips entries whose `map/status` is not `ACTIVE`.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

RULE	STATEMENT
Designated-position hold time	There is **no maximum hold time** on the designated-sponsor position. A designated MAP that receives no candidate remains designated until a candidate arrives. The rotation is event-driven — advance occurs on acceptance — so an idle period has no operational cost and no fairness implication.
`candidateRef`	`active[]/candidateRef` is **OPTIONAL**. When present, it carries a non-identifying reference assigned by the Sponsor MAP for its own coordination — for example a ticket number from the Sponsor's internal onboarding system. The field is free-form and is not standardised across MAPs; consumers outside the issuing MAP MUST NOT rely on its semantics.
`sponsorship.registrationURI`	`sponsorship.registrationURI` is **OPTIONAL**. When present, it advertises the Sponsor MAP's own onboarding application page (self-service portal, contracted form, or structured registration endpoint), enabling prospective MAPs and the TAG-published sponsor-registration page to deep-link into that MAP's onboarding flow. A MAP that uses non-electronic registration (email, telephone, ticketing) omits the field; prospective MAPs then use the MAP's `map/contact[]` entries as the contact channel.

5.5 Directory Section — Field Reference

JSON PATH	DESCRIPTION	TYPE	REQUIRED
`list[]`	Array of directory lists	Object Array	Yes
`list[]/listType`	List type (currently only `RCPID`)	String	Yes
`list[]/identity[]`	Array of CP identity objects	Object Array	Yes
`list[]/identity[]/id`	Unique RCPID (UUIDv4)	String	Yes
`list[]/identity[]/name`	Full brand name	String	Yes
`list[]/identity[]/processSupport[]`	Supported industry processes	Object Array	Yes
`list[]/identity[]/processSupport[]/process`	Process name: `OTS`, `GPLB`	String	Yes
`list[]/identity[]/processSupport[]/status`	Status: `ACTIVE`, `SUSPEND`, `TEST`	String	Yes
`list[]/identity[]/resource[]`	CP resources (URLs, contacts)	Object Array	Yes
`list[]/identity[]/resource[]/name`	Resource name (e.g., `customerAssistURL`)	String	Yes
`list[]/identity[]/resource[]/type`	Resource type: `url`, `email`, `phone`	String	Yes
`list[]/identity[]/resource[]/value`	Resource value	String	Yes
`list[]/identity[]/signing[]`	PKI signing configuration	Object Array	Yes
`list[]/identity[]/signing[]/domain`	Root domain for public key lookup	String	Yes
`list[]/identity[]/signing[]/routingID[]`	Routing IDs signed with this domain (regex)	String Array	Yes
`list[]/identity[]/contact[]`	Contact methods for this CP (same structure as `map/contact[]`)	Object Array	Optional

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

JSON PATH	DESCRIPTION	TYPE	REQUIRED
`list[]/identity[]/contact[]/type`	Contact type: `phone`, `email`, `url`	String (enum)	Yes
`list[]/identity[]/contact[]/value`	Contact value (number, address, or URL)	String	Yes
`list[]/identity[]/contact[]/purpose[]`	What the contact is for: `support`, `sales`, `technical`	String Array	Optional
`list[]/identity[]/contact[]/availabilityFrom`	Availability start time	Time (`HH:mm`)	Optional
`list[]/identity[]/contact[]/availabilityTo`	Availability end time	Time (`HH:mm`)	Optional
`list[]/identity[]/contact[]/availabilityDays[]`	Days available	String Array (day names)	Optional
`list[]/identity[]/contact[]/comment`	Usage notes	String	Optional

5.6 CP Contact Object

Each CP identity MAY include a `contact` array using the **same structure as the MAP contact object**. This provides CP-level contact details for operational purposes — typically CP-to-CP sales and support enquiries during the switching process.

Purpose

The CP contact object allows each CP to advertise its preferred contact methods, trading hours, and escalation points directly in the CP Registry. This eliminates the need for offline spreadsheets or specialised tools — a MAP's application can look up the other CP's preferred contact method in real time and present it to the agent handling the switch.

Usage Rules

- If `availabilityTo` and `availabilityFrom` are absent, the contact SHOULD be interpreted as **24-hour availability**
- If `availabilityDays` is absent, the contact SHOULD be assumed available **7 days a week**
- This encourages CPs to be always reachable and promotes optimal behaviour
- The `purpose` field allows granular control — a CP can advertise different contact methods for `sales`, `support`, and `technical` enquiries
- Contact details are held in the CP Registry and are only accessible via OAuth 2.0-authenticated API calls — they are **not publicly accessible** and must not be shared with end consumers

Relationship to MAP Contact Object

LEVEL	CONTACT OBJECT	TYPICAL USE
MAP (`map/contact[]`)	MAP operational contacts	MAP-to-MAP technical issues, delivery failures, infrastructure problems

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

LEVEL	CONTACT OBJECT	TYPICAL USE
CP (`list[]/identity[]/contact[]`)	CP operational contacts	CP-to-CP sales queries, Switch Match escalations, switch support

Both use the identical JSON structure (`type`, `value`, `purpose`, `availabilityFrom`, `availabilityTo`, `availabilityDays`, `comment`), making implementation straightforward — the same parsing code handles both.

Example Use Case

A CP agent has exhausted all local options trying to resolve a Switch Match failure. Using the MAP's application, the agent looks up the losing CP's contact object from the Master Registry. The application checks the current time and day, filters to the `sales` purpose, and presents the available contact methods:

- During working hours: phone and email are both available
- Outside working hours: only the email/ticketing URL is shown
- The agent raises a query through the preferred channel with relevant correlation IDs

This process requires no swivel-chair, no MAP involvement, and no specialised tools beyond the agent's normal interface.

5.7 TOTSCo Directory API

For MAPs that need to consume the TOTSCo directory (to route messages to CPs connected directly to the TOTSCo Hub), the TOTSCo Hub API Specification v2.0 defines the directory endpoint:

FIELD	VALUE
API Name	TOTSCo-DirectoryAPI
Context	`/directory`
Version	`v1`
HTTP Method	GET
Transport Security	HTTPS, TLS 1.3
Resource	`/entry`

Query format:
`https://{fqdn}/directory/{version}/entry?listType={listType}&identity={identity}`

PARAMETER	DESCRIPTION	REQUIRED
`listType`	Entity list type (e.g., `RCPID`)	Yes
`identity`	Specific RCPID, `all`, or omit for all entries	Optional

This directory can be used to update your local cache when you receive a message from an unknown source ID or encounter an unrecognised signing key.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

6. TOTSCo Integration

6.1 Overview

TOTSCo operates centralised hubs for both **One Touch Switch (OTS)** residential switching and **Switching for Business (SforB)**. Both hubs use the same **JAM message envelope** specification and the same **OAuth 2.0 over TLS 1.3** transport that the CSF uses, which gives the two networks strong compatibility at the message layer. For SforB, TOTSCo can participate in the CSF as a **Hub Managed Access Provider (HMAP)** — routing messages between its connected CPs and the wider CSF network.

This chapter describes how the CSF and TOTSCo interoperate. The detailed implementation plan has been shared separately with TOTSCo and is available on request from the TAG.

> CP MIGRATION PATH (SFORB AND NON-OTS ONLY).
> A CP currently using TOTSCo for **SforB** (and / or for niche bilateral exchanges or other non-OTS protocols) who wishes to **become a MAP of 1 on the CSF** for that traffic is covered by Part 1 §7.4 *MAP-of-1 Onboarding (incl. TOTSCo → CSF Migration)*. The CSF treats this as a first-class adoption path; the strong JAM-envelope compatibility makes the move a transport switch rather than a business change. **OTS is out of scope** — OTS is Ofcom-mandated to use TOTSCo as the single industry platform, so a CP that operates OTS MUST continue to use TOTSCo for OTS traffic. The migration applies to SforB and other non-OTS protocols only.

6.2 Integration Principles

- **Zero to low impact** on existing TOTSCo infrastructure
- The CSF uses the **same JAM message specification** and **OAuth 2.0 authentication** as TOTSCo
- No modifications to the TOTSCo letterbox API are required
- TOTSCo's existing OTS connections with MAPs can be reused

6.3 Registering CSF Endpoints

Impact: Zero to Low

It is highly probable that existing MAPs already have connectivity with TOTSCo from OTS or SforB. Any new MAP or MAP-of-1 joining the CSF must also register with TOTSCo following their existing procedures:

- Provide IP addresses for allowlisting
- Exchange OAuth 2.0 or API key credentials
- Confirm message exchange conforms to the JAM Specification

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

The only additional requirement is a **Directory API endpoint URL** for TOTSCo to retrieve the MAP's CSF registry.

6.3.1 Dedicated, locked-down HMAP interfaces — both directions

The CSF and TOTSCo deliberately adopt a **per-partner, dedicated-endpoint** operating model in both directions. This mirrors how TOTSCo already runs OTS and SforB today — every CP and MAP that connects to the TOTSCo Hub does so through an endpoint TOTSCo provisions specifically for that partner.

DIRECTION	WHAT THE MAP DOES	WHAT TOTSCO DOES
MAP-side (inbound to the MAP from TOTSCo)	Each CSF MAP exposes a **dedicated HMAP interface exclusively for TOTSCo**. This interface is not used by any other MAP or HMAP. It is **locked down** at the agreed contract — CSF spec increments, feature releases, envelope changes, and routing-group additions do **not** propagate through this interface. TOTSCo's exchanges with the MAP continue to work exactly as they did on day one.	TOTSCo treats the MAP's HMAP interface as a stable bilateral contract.
TOTSCo-side (inbound to TOTSCo from the MAP)	The CSF MAP delivers SforB messages to a dedicated TOTSCo letterbox URL provisioned specifically for that MAP.	TOTSCo provisions **one dedicated CSF letterbox URL per CSF MAP**, the same per-partner pattern TOTSCo already uses for every OTS / SforB partner today. Each URL is backed by the same hub infrastructure; only the URL path differs.

> WHY LOCKED DOWN.
> The architectural commitment to TOTSCo is that the HMAP interface is **steady-state**. The CSF can — and will — continue to evolve at its own cadence; that evolution happens behind each MAP's standard CSF interface, not through the dedicated HMAP interface TOTSCo consumes. This keeps TOTSCo's integration a one-time build, not an evergreen one, and isolates TOTSCo's regulated OTS operations from any CSF iteration.

> BEST PRACTICE (LEGACY HEADING RETAINED FOR BACKWARD COMPATIBILITY OF CROSS-REFERENCES).
> All MAPs should configure a dedicated, locked-down HMAP interface for TOTSCo. This is mandatory in the operating model described above and is the only pattern this section supports.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

6.4 Directory Collection

Impact: Medium

TOTSCo must collect the CP Registry from each MAP on the CSF network. This involves:

1. **Accessing each MAP's CP Registry endpoint** using existing OAuth 2.0 credentials
2. **Converging the collected CP Registries** into a Master Registry — either merged with the existing TOTSCo directory or maintained separately
3. **Updating periodically** — every 5 minutes, hourly, or daily, at TOTSCo's discretion

The CSF CP Registry is intentionally a **strict superset** of the TOTSCo directory structure for OTS and SforB, making convergence straightforward. TOTSCo can choose to retain all data, or use the CSF-specific fields (`signing[]`, expanded `resource[]`, `contact[]`, `outage[]`) for the value they add, or ignore them entirely — none of those choices break TOTSCo's existing parser.

Dual-process directory in TOTSCo today. TOTSCo's pre-production Directory v2 already serves both OTS and GPLB (SforB) entries from a single list, distinguishing them by `processSupport[].process`. MAPs that want a single-process view filter on an `identity` query parameter:

```
https://preprod.otshub.totsco.co.uk/directory/v2/entry?listType=RCPID&identity=all
https://preprod.otshub.totsco.co.uk/directory/v2/entry?listType=RCPID&identity=ots
https://preprod.otshub.totsco.co.uk/directory/v2/entry?listType=RCPID&identity=gplb
```

The production model is expected to follow the same shape. The CSF's Master Registry follows the same convention (`processSupport[].process` carries `OTS`, `GPLB`, and future process tags), so TOTSCo can apply its existing filter logic unchanged.

The number of CSF MAPs is anticipated not to exceed ten initially.

6.5 RCPID Format Handling

Impact: Low

The SforB Message Specification defines source and destination identity fields as strings (up to 256 characters). The CSF uses UUIDv4 (36 characters) — well within this constraint.

In practice:

- **CSF MAP sending to TOTSCo**: Uses the destination RCPID from the TOTSCo directory (typically 4-character OTS format)
- **TOTSCo sending to CSF MAP**: Uses the destination RCPID from its Master Registry (UUIDv4 format, collected from CSF CP Registries)

TOTSCo automatically acquires all CP UUIDs when collecting MAP registries. Both CSF and TOTSCo already accept UUIDv4 for correlation IDs.

6.6 API Compatibility

Impact: Zero

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

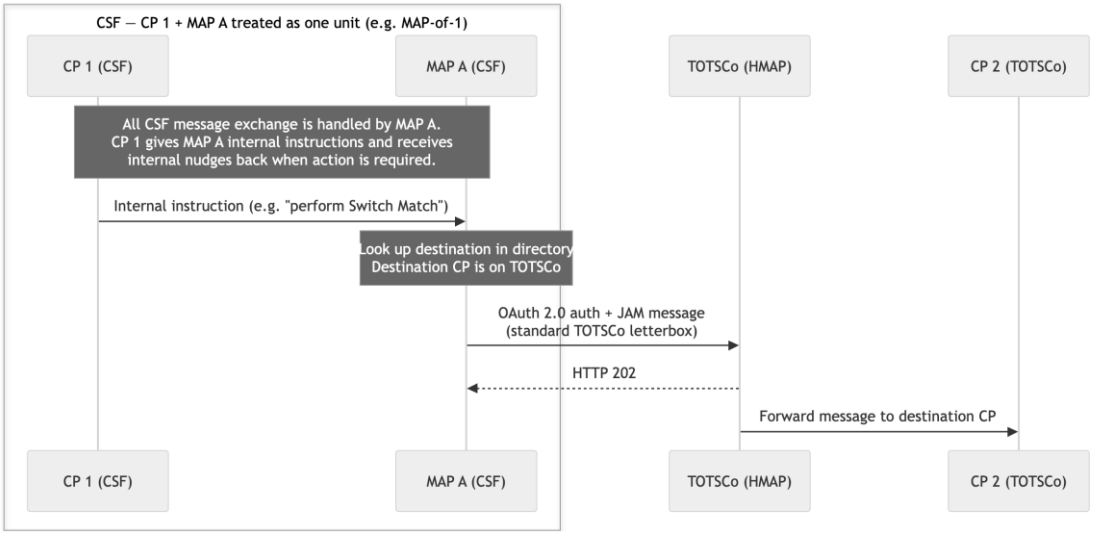
TOTSCo uses the industry JAM Specification for message exchange — the same standard used by the CSF. The CSF validates the JSON envelope identically and uses OAuth 2.0 over TLS 1.3.

ASPECT	TOTSCO	CSF	COMPATIBLE?
Message format	JAM Specification	JAM Specification	Yes
Envelope validation	Same rules	Same rules	Yes
Authentication	OAuth 2.0	OAuth 2.0	Yes
Transport	TLS 1.3	TLS 1.3	Yes
Success response	HTTP 202	HTTP 202	Yes
Error codes (sync)	9xxx	9xxx + 8xxx (PKI only)	Yes

The 8xxx PKI error codes are CSF-specific and are not sent to TOTSCo. No API modifications are required on the TOTSCo side.

6.7 Message Flow: CSF to TOTSCo

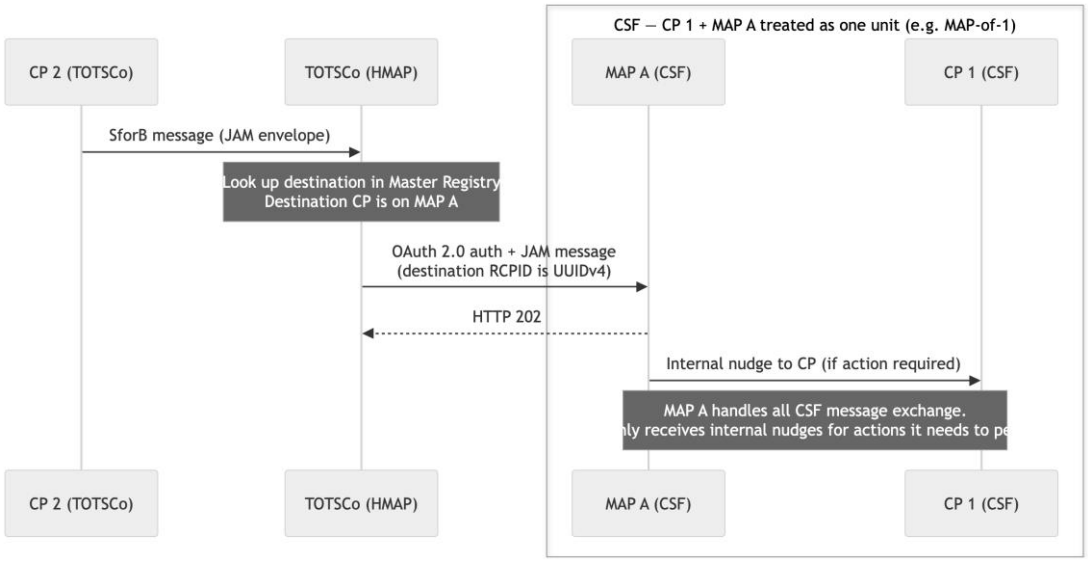
In CSF, **CP 1 and MAP A are treated as a single unit at the transport layer** — the MAP handles all CSF message exchange, and the CP only gives the MAP internal instructions and receives internal nudges back when action is required. This is true whether the CP is operating as a **MAP-of-1** (CP and MAP are literally the same entity) or is a CP being served by a separate MAP. The mermaid diagram below uses a grouping box to make this visible.



© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

6.8 Message Flow: TOTSCo to CSF

The same CSF grouping applies on the inbound side — MAP A receives the message from TOTSCo on behalf of CP 1, and CP 1 is notified internally if the message requires action.



6.9 PKI Considerations

TOTSCo, as an HMAP, is **not required** to implement DKIM/PKI signing when exchanging messages on behalf of its CPs. This is by design — to minimise integration cost for hub operators.

However:

- TOTSCo **is strongly recommended** to implement the full CSF CP Registry format for SforB, as its CP clients benefit significantly from the additional contact and support information
- TOTSCo **is not precluded** from implementing PKI if desired — the CSF framework supports it
- When a CSF MAP receives a message from TOTSCo's dedicated endpoint, it knows the message comes from a trusted hub and applies hub-specific validation rules rather than PKI verification

6.10 Passthrough Headers

When TOTSCo forwards a message from a CSF MAP to its own CPs, the `X-CSF-SIGNATURE` header SHOULD be left intact. This provides end-to-end trust verification — the destination CP (or its MAP) can verify the original signing even after hub forwarding.

6.11 Support and Diagnostics

By accessing CSF registries, TOTSCo gains direct access to MAP contact details (phone, email, URL with availability hours), enabling prompt issue resolution.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

The CSF registry provides more detailed contact information than currently available through OTS, including:

- MAP and CP support hours
- Multiple contact methods with purpose tags (sales, support, technical)
- Service status URLs for real-time operational monitoring

6.12 Further Information

DOCUMENT	DESCRIPTION
TOTSCo HMAP Integration Guide	The **TOTSCo-side build-and-commission handbook**: executive summary for middle management, six concrete deliverables (Directory API client, Master Registry convergence, RCPID translation, routing rules, dedicated CSF letterbox URL, optional PKI), five-phase plan (Discovery → Build → Test → Operational Commissioning → Production), copy-and-run test patterns, cost-and-effort estimate, risk register, and collective industry-benefit framing. Companion to this chapter; **read this guide alongside §6**.
TOTSCo Integration — Implementation Plan v1.1	Detailed integration steps (shared with TOTSCo)
TOTSCo Integration — Overview Slides v1.0	High-level integration overview
TOTSCo Hub API Specification v2.0	Industry-standard letterbox and directory API specification for SforB

These documents are available on request from the TAG.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

7. CP Transitions

7.1 Overview

CPs are free to transition between MAPs, subject to standard commercial terms. The CSF is designed to make this transition seamless, with built-in mechanisms for identity portability, in-flight order transfer, and conflict resolution.

A CP can operate through only **one MAP at any given time** for a given RCPID. This relationship is established through DNS records controlled by the CP.

7.2 Reasons for Moving

A CP may move to a new MAP for various reasons:

- **Service dissatisfaction**: The existing MAP fails to meet agreed standards
- **Provider termination**: The current MAP enters administration or issues a termination notice
- **Strategic realignment**: The CP's business strategy requires a different MAP
- **Cost optimisation**: The CP wishes to become a MAP of 1 or consolidate providers
- **M&A** - A CP is purchased by another CP and their brand rolls into that of the acquirer

7.3 Transition Process

Step 1 — Establish with New MAP

The CP establishes a commercial relationship with the new MAP and completes the CP onboarding process:

- New PKI key pair is generated by the new MAP
- The new MAP does NOT yet add the CP to its directory

Step 2 — DNS Update

The CP updates its DNS TXT records. CPs should lower their TTL on the existing records in the days/hours prior:

- `[RCPID]._domainkey.[cp-domain]` — new public key from the new MAP
- `[RCPID]._mapkey.[cp-domain]` — URL of the new MAP's message endpoint

> NOTE
> : This temporarily stops message flow via the original MAP. Messages sent by other MAPs will fail PKI verification against the old MAP and be queued locally for retry.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Step 3 — Directory Update

The new MAP performs the directory-side activation steps that make the CP visible and routable through it:

1. **Add the CP entry** to the new MAP's CP Registry under `list[].identity[]`, with the agreed `processSupport[]` array covering each SforB process the CP is signing up for (`businessSwitchMatch`, `businessSwitchOrder`, `businessSwitchOrderUpdate`, `businessSwitchOrderTrigger`, `businessSwitchOrderCancellation`, etc).
2. **Set per-CP status.** During cut-over the entry SHOULD be published with `processSupport[].status = TEST` if the new MAP is still onboarding the CP; or `ACTIVE` if the new MAP itself is already an `ACTIVE` participant and the only change is the move. The status is later transitioned to `ACTIVE` in Step 6, after in-flight reconciliation has cleared.
3. **Bump the registry `version` timestamp** so other MAPs polling the directory observe the change on their next refresh.
4. **Re-sign and re-publish** the new MAP's CP Registry per Directory API §5. Do NOT attempt to suppress the entry that still exists in the original MAP's registry — until Step 5 completes there is a deliberate, brief overlap which the conflict-resolution rules in §7.5 are designed to absorb.
5. **Wait for cache propagation.** Receiving MAPs cache registry data; allow at least one configured polling cycle (typically a few minutes) before relying on the entry being visible network-wide. The new MAP's registry endpoint MUST honour `If-Modified-Since` so peers can confirm freshness without polling the full payload.
6. **DNS already settled.** DNS was updated in Step 2; the directory entry is the authoritative *advertisement* of the routing change but the underlying authority comes from `_mapkey`, which receivers will re-resolve before sending sensitive traffic.

> NOTE
> : Until Step 5 has run and the original MAP's registry no longer lists the CP, the CP appears under two MAPs simultaneously. This is the expected transient state and is what triggers the conflict-detection rules in §7.5.

Step 4 — RCPID Status Export Request

The new MAP sends a signed **`rcpidStatusRequest`** to the original MAP via the Letterbox API. This request is the formal handover trigger and authorises the original MAP to release in-flight order data and decommission its registry entry for the CP.

Authorisation model. Before responding, the original MAP MUST independently verify all three of:

1. The DKIM signature on the inbound request is valid per PKI §3 and the signing CP's `_domainkey` resolves to the new MAP.
2. The moving CP's `_mapkey` DNS record now resolves to the *requesting* MAP. This is the cryptographic proof that the CP authorised the move — DNS is controlled by the CP, so a `_mapkey` pointing at the requesting MAP can only have been set with the CP's consent.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

3. The requesting MAP's registry advertises the CP under it (at any status — `TEST` or `ACTIVE`).

Request payload.

```
{
  "envelope": {
    "source":      { "type": "RCPID", "identity": "<new-MAP RCPID>",  "correlationID": "<uuid>" },
    "destination": { "type": "RCPID", "identity": "<old-MAP RCPID>" },
    "routingID":   "rcpidStatusRequest",
    "auditData":   [ { "name": "test", "value": "false" } ]
  },
  "rcpidStatusRequest": {
    "movingRCPID":   "<the moving CP's RCPID>",
    "purpose":       "MAP_TRANSFER",
    "newMAP":        "<new-MAP RCPID>",
    "newMAPDomain":  "letterbox.newmap.example",
    "requestedAt":   "2026-04-29T10:00:00Z",
    "specVersion":   "GPLB-SG SforB Msg Spec v3.0"
  }
}
```

Response time and failure modes.

CONDITION	RESPONSE	HTTP / CODE
All authorisation checks pass	Synchronous `rcpidStatusConfirmation` carrying the export (see §7.4)	202
DNS `_mapkey` does not resolve to the requester	Reject — handover not authorised	403 / `9301`
DKIM verification fails	Reject — signature invalid	403 / `8101`
No in-flight SORs found for the moving CP	`rcpidStatusConfirmation` with empty `orders[]` array; registry entry still removed	202
Original MAP cannot comply within SLA	`rcpidStatusFailure` with retry guidance	503 / `9302`

SLA expectations. The original MAP SHOULD respond within 30 seconds during core business working hours for an export of fewer than 100 in-flight SORs and within 2 minutes for a complete export. A delayed response forces the new MAP to retry; the original MAP MUST handle idempotency by correlating retries against the original `correlationID`.

Side effects on the original MAP. As soon as the original MAP commits to processing the export, it MUST:

- **Quiesce the moving CP's mailbox** — block any new outbound SforB messages from the CP and queue inbound messages addressed to it until the response has been sealed, so that the export captures a consistent snapshot.
- **Remove the CP entry** from its CP Registry and re-sign and re-publish the registry version (so that other MAPs can stop attempting to route to it).
- **Persist the export** alongside the inbound `rcpidStatusRequest` correlation ID for at least 30 days for audit, regardless of the response outcome.

Step 5 — In-Flight Order Reconciliation

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Step 5 is where the in-flight Switch Orders move from the original MAP's state engine into the new MAP's state engine, and where every counterparty MAP gets the cross-check it needs to keep treating the customer journey as continuous.

7.3.5.1 Suggested Export Routine (run by the original MAP)

The export routine runs server-side on the original MAP between accepting the `rcpidStatusRequest` and emitting the response. Each step is idempotent so the routine can be safely re-entered after a partial failure:

1. **Hold the moving CP's mailbox.** Block new outbound SforB messages from the moving CP (the GRCP path) and queue any inbound SforB messages addressed to it (the LRCP path). This freeze prevents new in-flight state from being created during the export window.
2. **Enumerate SORs.** Walk the SforB order store for every `switchOrderReference` where the moving CP is either the GRCP (originator of the match/order) or the LRCP (counterparty). Each SOR is processed independently.
3. **Capture the last completed interaction only.** For each SOR, identify the most recent completed request/response pair (e.g. `businessSwitchOrderRequest` → `businessSwitchOrderConfirmation`) per service. Partially completed exchanges are dropped, per the existing rule in §7.4.5.
4. **Resolve counterparty routing.** For every SOR, record the counterparty CP's RCPID and that CP's *current* MAP — read live from the CP Registry, not from the cached SOR record — so the new MAP knows where to direct follow-up `rcpidStatusRequest` cross-checks in the reconciliation phase.
5. **Snapshot per-service state.** For every service inside the SOR (`IAS`, `NBICS`, `NBICSRange`, etc.) capture the current `status` from the SforB state machine (`matched`, `ordered`, `triggered`, `cancelled`, `failed`), the active `plannedSwitchDate`, and any `fault` block recorded against the service. This mirrors the structure documented in the GPLB-SG SforB Message Specification v3.0 §3.3 — Business Switch Order Request (canonical source published by GPLB-SG via the FCS).
6. **Capture original-match identity context.** Carry forward the customer-identifying fields (`grcpBrandName`, `name`, `account`, `identity[]`, `address`) from the `businessSwitchMatchRequest/Confirmation` that opened the SOR. This allows the new MAP to validate continuity if the counterparty asks for a re-match.
7. **Compute integrity hashes.** Canonicalise each SOR record and compute a SHA-256 `recordHash`, then an `aggregateHash` over the whole `orders[]` array. Sign `aggregateHash` with the original MAP's `_domainkey` private key (the MAP-level key from PKI §3, NOT the CP-level key being moved).
8. **Wrap and return.** Emit the response payload described in §7.4 as the body of the `rcpidStatusConfirmation` message.
9. **Finalise registry removal.** Once the response has been transmitted, finalise removal of the CP from the registry as already initiated in Step 4 and re-publish.

7.3.5.2 Reconciliation Performed by the New MAP

After receiving the export the new MAP:

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

1. **Verifies the aggregate signature** against the original MAP's `_domainkey` — this establishes chain-of-custody for the data being imported.
2. **Imports each SOR** into its own switching state engine, **preserving the `switchOrderReference` UUID** so the customer journey is not restarted. New correlation IDs are minted at the envelope layer for any subsequent outbound traffic, per the §7.4.5 rule.
3. **Issues an `rcpidStatusRequest` to each counterparty MAP** named in the export — one per counterparty — to:

    - Notify them that the moving CP is now reachable via the new MAP.
    - Cross-check the SOR state from the counterparty's perspective. Any mismatch (e.g. counterparty believes a service is `triggered` when the export says `ordered`) is flagged and resolved bilaterally before further messages flow.

1. **Resumes message flow** by re-publishing the CP entry with `processSupport[].status = ACTIVE` in the new MAP's CP Registry once all reconciliation responses are received and clean.
2. **Notifies the moving CP** with a transition report (count of SORs imported, any reconciliation flags raised, and estimated time to full resumption).

7.3.5.3 What Must Be in the Export — Minimum Detail per SOR

The full schema is documented in §7.4. At minimum, every in-flight SOR record MUST carry:

FIELD GROUP	PURPOSE
`switchOrderReference`, `role`, `currentState`	Identity of the order and which side the moving CP is on
`counterparty.{rcpid, mapRcpid}`	Tells the new MAP where to send the cross-check request
`originalMatch.{grcpBrandName, name, account, identity[], address}`	Identity context preserved from the opening match
`services[].{serviceType, serviceIdentifier, status, plannedSwitchDate, fault}`	Authoritative per-service state at the moment of export
`lastInteraction.{routingID, timestamp, sourceCorrelationID, destCorrelationID, payloadHash}`	The most recent completed exchange — the resumption baseline
`expectedNext.{messageType, deadline, slaWindow}`	What message is due next and the SLA clock the new MAP must continue to honour
`auditTrail[]`	Hashes (not bodies) of every prior completed exchange so anomalies can be investigated bilaterally without exposing message body content

Step 6 — Resume Operations

With the export imported and counterparty cross-checks clean, the new MAP can promote the moving CP to fully operational state. Treat the lists below as a sign-off checklist; nothing in Step 6 can be skipped without recording a deviation.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

New MAP

- [ ] Cross-check responses received from every counterparty MAP referenced in the export, and any mismatches resolved bilaterally before resuming traffic.
- [ ] Per-CP `processSupport[].status` flipped from `TEST` to `ACTIVE` in the registry; registry version bumped, re-signed and re-published.
- [ ] Synthetic round-trip executed against the moving CP via the TAG Baseline Test Suite — §8.5 Letterbox API to confirm end-to-end delivery.
- [ ] Audit log entry written: time of completion, count of SORs imported, `aggregateHash` of received export, original MAP RCPID, signing key fingerprint.
- [ ] Telemetry resumed: outbound messages no longer carry `auditData.test=true` (unless the new MAP is itself still in `TEST` per Onboarding & Testing Process §5).
- [ ] CP-facing portal or dashboard reflects every imported SOR with continuity preserved (same `switchOrderReference`, same per-service `status`).

Original MAP

- [ ] CP entry removed from registry; registry version bumped, re-signed and re-published.
- [ ] `_domainkey` retained for the agreed retention window so receivers verifying historic signatures within that window MUST still be able to resolve them.
- [ ] Inbound queue for the moving CP drained: each remaining queued message either forwarded under cover of an `rcpidStatusRequest` to the new MAP, or rejected with code `9999` (in migration) if forwarding is not feasible.
- [ ] Export package archived alongside the inbound `rcpidStatusRequest` correlation ID for at least 30 days.
- [ ] Operational support for the moving CP closed off cleanly, with the agreed handover support window communicated to the CP and the new MAP.

Counterparty MAPs

- [ ] Registry caches refreshed; the new MAP entry observed for the moving CP.
- [ ] `_mapkey` DNS resolved at least once after the directory change to confirm the authoritative MAP.
- [ ] Locally queued messages drained — those that were retrying against the original MAP MUST be re-signed and re-routed to the new MAP's letterbox.
- [ ] Bilateral SOR cross-check with the new MAP reported clean (or anomalies escalated through the conflict-resolution path in §7.5).

The CP

- [ ] DNS records propagated globally — verify from at least three geographic resolvers before declaring the migration complete.
- [ ] All in-flight customer journeys visible in the new MAP's portal/dashboards.
- [ ] Old MAP support contact still reachable for the agreed handover support window.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- [ ] Stakeholder communications sent: customers, downstream CPs, and any affected industry forum.

7.4 RCPID Status Export Schema

The schema below is a CSF design proposal layered onto the GPLB-SG SforB Message Specification v3.0 to support lossless MAP migration. It is published here so that the TAG can ratify the field set and the canonicalisation rules used for `aggregateHash`/`recordHash` before the first MAP migration runs in production.

7.4.1 Top-Level Envelope

The export rides inside the standard CSF envelope. The body element is `rcpidStatusConfirmation` and contains two top-level objects: `exportMetadata` and `orders[]`.

```
{
  "envelope": { "...": "as per Letterbox §4.3" },
  "rcpidStatusConfirmation": {
    "exportMetadata": { "...": "see §7.4.2" },
    "orders":         [ { "...": "see §7.4.3" } ]
  }
}
```

7.4.2 Export Metadata

FIELD	DESCRIPTION	TYPE	REQUIRED
`exportVersion`	Schema version of this export. Current: `1.0`.	String	Yes
`generatedAt`	UTC timestamp at which the export was sealed.	String (ISO 8601)	Yes
`generatedBy`	RCPID of the original (sending) MAP.	String	Yes
`movingRCPID`	RCPID of the CP being transferred.	String	Yes
`newMAP`	RCPID of the new MAP — echoes the requester.	String	Yes
`dnsVerifiedAt`	UTC timestamp at which the original MAP confirmed `_mapkey` resolution to the new MAP.	String	Yes
`directoryRemovedAt`	UTC timestamp at which the moving CP was removed from the original MAP's registry.	String	Yes
`specReference`	Version of the SforB Message Spec the orders were created against (e.g. `GPLB-SG SforB Msg Spec v3.0`).	String	Yes
`orderCount`	Number of records in `orders[]`.	Integer	Yes
`aggregateHash`	SHA-256 hash over the canonicalised `orders[]` array.	String (hex)	Yes
`signature`	Base64 signature over `aggregateHash` using the original MAP's `_domainkey` private key (algorithm `rsa-sha256` or `ed25519-sha256` per PKI §3).	String	Yes

7.4.3 Per-Order Record

Each entry in `orders[]` is a full snapshot of a single in-flight Switch Order. The schema is designed to map 1:1 onto the SforB state machine in the GPLB-SG SforB Message Specification v3.0 §3 so the new MAP can resume handling without re-deriving state.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

FIELD	DESCRIPTION	TYPE	REQUIRED
`switchOrderReference`	UUID of the SOR. Preserved verbatim — the new MAP MUST NOT mint a new SOR.	String	Yes
`role`	`gaining` if the moving CP is the GRCP for this SOR; `losing` if it is the LRCP.	String (enum)	Yes
`currentState`	High-level state: `matched`, `ordered`, `partially-triggered`, `partially-cancelled`, `complete-pending-trigger`, `trigger-failed`.	String (enum)	Yes
`counterparty.rcpid`	RCPID of the CP on the opposite side of the order.	String	Yes
`counterparty.mapRcpid`	RCPID of the counterparty's *current* MAP, looked up live at export time.	String	Yes
`originalMatch.grcpBrandName`	Brand name presented to the customer at match time.	String	If known
`originalMatch.name`	Customer or business name from the match.	String	Yes
`originalMatch.account`	LRCP-side account number, if supplied at match.	String	Optional
`originalMatch.identity[]`	Identity tuples (`contact`, `IAS`, `NBICS`, `SecureID`, `SIC`) used to anchor the match.	Array	Optional
`originalMatch.address`	UPRN / address lines / post town / post code from the match.	Object	Yes
`services[].serviceType`	`IAS`, `NBICS`, `NBICSRange` (or future industry-agreed type).	String	Yes
`services[].serviceIdentifier`	The GRCP's per-service reference. Required for every service, mirroring §3.3 of the SforB Msg Spec.	String	Yes
`services[].status`	Current service state: `matched`, `ordered`, `triggered`, `cancelled`, `failed`.	String	Yes
`services[].plannedSwitchDate`	Per-service planned switch date if it overrides the SOR header.	String (CCYY-MM-DD)	Optional
`services[].fault`	`{ faultCode, faultText }` if the service is in `failed` state.	Object	Optional
`lastInteraction.routingID`	Most recent completed exchange (e.g. `businessSwitchOrderConfirmation`).	String	Yes
`lastInteraction.timestamp`	When that exchange completed.	String (ISO 8601)	Yes
`lastInteraction.sourceCorrelationID`	The original sender-side correlation ID.	String	Yes
`lastInteraction.destCorrelationID`	The destination-side correlation ID, if available.	String	Optional
`lastInteraction.payloadHash`	SHA-256 of the canonicalised message body — included so the new MAP can verify with the counterparty without exposing the body itself.	String (hex)	Yes
`expectedNext.messageType`	The message type expected next on this SOR (e.g. `businessSwitchOrderTriggerRequest`).	String	Optional

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

FIELD	DESCRIPTION	TYPE	REQUIRED
`expectedNext.deadline`	Hard SLA deadline by which the next message must be sent.	String (ISO 8601)	Optional
`expectedNext.slaWindow`	The SLA budget remaining as ISO 8601 duration.	String	Optional
`auditTrail[]`	Array of `{ routingID, timestamp, payloadHash }` for every prior completed exchange on this SOR. Hashes only — message bodies are NOT exported.	Array	Yes
`recordHash`	SHA-256 over the canonicalised record above (excluding `recordHash` itself). Allows individual orders to be re-verified after import.	String (hex)	Yes

7.4.4 Worked Example

A minimal export covering one SOR where the moving CP is the GRCP and the order is in the `ordered` state, awaiting trigger:


```
{
  "rcpidStatusConfirmation": {
    "exportMetadata": {
      "exportVersion": "1.0",
      "generatedAt": "2026-04-29T10:00:00Z",
      "generatedBy": "<old-MAP RCPID>",
      "movingRCPID": "809b6e65-a6e7-40f6-8b52-04dd65b6fce1",
      "newMAP": "<new-MAP RCPID>",
      "dnsVerifiedAt": "2026-04-29T09:58:42Z",
      "directoryRemovedAt": "2026-04-29T09:59:01Z",
      "specReference": "GPLB-SG SforB Msg Spec v3.0",
      "orderCount": 1,
      "aggregateHash": "5a3b...c812",
      "signature": "MEUCIQDh..."
    },
    "orders": [
      {
        "switchOrderReference": "123e4567-e89b-12d3-a456-426614174000",
        "role": "gaining",
        "currentState": "ordered",
        "counterparty": {
          "rcpid": "d8322d80-92c8-4906-a408-f1d2daf7e03d",
          "mapRcpid": "<counterparty MAP RCPID>"
        },
        "originalMatch": {
          "grcpBrandName": "Bridgend Telecoms (Telesales)",
          "name": "The Coffee Shop",
          "account": "0003316563216",
          "identity": [
            { "name": "contact", "value": "John Smith" },
            { "name": "NBICS",   "value": "01234987654" },
            { "name": "SIC",     "value": "AB1234" }
          ],
          "address": {
            "uprn": "12345",
            "addressLines": ["Unit 1", "Highlands Industrial Estate", "22 Cheshunt Mews"],
            "postTown": "Glasgow",
            "postCode": "SW2R 5AF"
          }
        },
        "services": [
          { "serviceType": "IAS",        "serviceIdentifier": "ABC123",                "status": "ordered" },
          { "serviceType": "NBICS",      "serviceIdentifier": "0101111222",            "status": "ordered" },
          { "serviceType": "NBICSRange", "serviceIdentifier": "0101112000-0101112999", "status": "ordered",
"plannedSwitchDate": "2025-05-27" }
        ],
        "lastInteraction": {
          "routingID": "businessSwitchOrderConfirmation",
          "timestamp": "2026-04-12T14:21:08Z",
          "sourceCorrelationID": "ABC123",
          "destCorrelationID": "XYZ987",
          "payloadHash": "8f1a...0e44"
        },
        "expectedNext": {
          "messageType": "businessSwitchOrderTriggerRequest",
          "deadline": "2025-04-03T23:59:59Z",
          "slaWindow": "P14D"
```


© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
        },
        "auditTrail": [
          { "routingID": "businessSwitchMatchRequest",      "timestamp": "2026-04-08T09:00:00Z", "payloadHash": "..." },
          { "routingID": "businessSwitchMatchConfirmation", "timestamp": "2026-04-08T09:00:18Z", "payloadHash": "..." },
          { "routingID": "businessSwitchOrderRequest",      "timestamp": "2026-04-12T14:20:55Z", "payloadHash": "..." },
          { "routingID": "businessSwitchOrderConfirmation", "timestamp": "2026-04-12T14:21:08Z", "payloadHash": "..." }
        ],
        "recordHash": "9c2d...4e10"
      }
    ]
  }
}
```

7.4.5 Rules

- **Switch Orders only.** The export includes Switch Orders only. Switch Matches that have not progressed to an order MUST be repeated by the CP via its new MAP.
- **Completed interactions only.** Partially completed interactions MUST NOT be exported — only the last completed interaction for each exchange.
- **Correlation IDs.** The new MAP mints fresh envelope correlation IDs for any subsequent messages — the originals are recorded in `lastInteraction` and `auditTrail[]` for traceability only.
- **Directory update precedence.** The original MAP MUST remove the moving CP from its directory before transmitting the export response. Receivers can rely on the registry version bumping in tandem with the export.
- **Body hashes, not bodies.** Only message-body hashes are exported. The body itself is opaque to the CSF and remains the property of the originating CP — the hashes are sufficient to support bilateral counterparty cross-checks without re-exposing customer data.
- **Signed end-to-end.** The export's `aggregateHash` MUST be signed with the original MAP's `_domainkey` private key (the MAP-level key), giving the new MAP and any audit body cryptographic proof of provenance for the imported data.

7.5 Conflict Resolution

When a CP appears in more than one MAP's directory simultaneously:

1. The detecting MAP checks the CP's `_mapkey` DNS record
2. DNS indicates the current MAP endpoint
3. The MAP marks the conflicting entry as being in "conflict"
4. The new MAP advertises the old MAP as being in conflict, signalling other MAPs to check

Over time, all MAPs will recognise the conflict and update their directories accordingly.

Latency During Transition

A CP should plan its move to minimise the gap — typically done overnight. During the transition:

- Messages sent by other MAPs will reflect routing from their current directories

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- PKI verification will eventually prevent delivery to the old MAP
- Messages will be queued locally by sending MAPs for retry
- Once the new MAP has completed DNS changes and RCPID Status exports, messages resume flowing to the correct MAP

This self-healing methodology is native to the distributed CP Registry model — PKI provides the final verification check before sending.

7.6 RACI for In-Flight Order Handling

ACTIVITY	ORIGINAL MAP	NEW MAP	CP
Generate new PKI keys	—	Responsible	—
Update DNS records	—	Assists	Responsible
Stop processing messages (after DNS change)	Automatic	—	—
Respond to RCPID Status request	Responsible	—	—
Remove CP from directory	Responsible	—	—
Add CP to directory	—	Responsible	—
Send RCPID Status requests to other MAPs	—	Responsible	—
Import in-flight order data	—	Responsible	—
Notify affected CPs	—	Responsible	Informed

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

8. Commercial Scenarios

8.1 Overview

The CSF addresses several commercial scenarios that were not fully covered by OTS. These scenarios are critical for ensuring business continuity, CP autonomy, and consumer/business protection.

8.2 CP Emergency Migration

CP Emergency Migration is the reactive process by which a CP transfers to a new MAP under one of the §8.2.2 trigger conditions (most often a failing, suspended, removed, or unresponsive MAP). It contrasts with a *normal* CP transition (§7), which is planned and CP-driven on a commercial timeline. The CSF mandates that every MAP supports emergency on-boarding of an existing CP as a core capability.

8.2.1 Definition and Scope

Emergency migration applies when the CP needs to move with little or no notice — usually because the current MAP can no longer be relied on to sign, route, or audit the CP's messages. It is *not* a substitute for a normal transition (§7); it is the framework's safety net.

Under OTS, a CP cannot move its in-flight orders between MAPs without significant manual effort, and the same risk exists if a MAP enters administration. The CSF addresses this by:

- Decoupling the CP from the MAP through DNS-controlled identity (§6)
- Enabling CPs to move freely between MAPs (or across multiple MAPs under different brands)
- Mandating that every MAP supports in-flight order export as a core capability (§4)
- Providing the `rcpidStatusRequest` flow so a new MAP can reconstruct or import in-flight state from the old MAP (§4) or, when the old MAP is offline, from other MAPs holding the orders

8.2.2 Triggers

Any one of the following invokes CP Emergency Migration:

#	TRIGGER	SOURCE / AUTHORITY
T1	MAP failure / outage (technical, not by TAG decision)	Operational
T2	MAP enters administration or insolvency	Statutory / commercial
T3	MAP **suspended** by TAG (after governance review)	§10 governance — `SUSPENDED` directory status
T4	MAP **removed** by TAG (after 30-day remediation window)	§10 governance — removal from Master Registry
T5	MAP unresponsive (operationally unreachable but not formally suspended)	CP / TAG-observable behaviour — repeated `rcpidStatusRequest` non-response, `9301` reject

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

#	TRIGGER	SOURCE / AUTHORITY
		noise above threshold, sustained failure to advertise in directory
T6	CP-initiated emergency (CP invokes emergency rather than normal transition — e.g. MAP material breach of MoU / SLA)	CP–MAP commercial contract

Triggers T1, T2, T5 cover the "MAP can't do its job" cases. T3 and T4 are TAG-driven and are also referenced in the **draft Anti-Competitive Enforcement clause** — to be published as an **Appendix to the CSF framework** once it has been legally reviewed and signed off by the TAG members (tracked as P-3 in the open-conflicts log). Until the Appendix is published, T3/T4 follow the existing §10.7 graduated enforcement ladder. T6 covers the rare case where the CP itself elects to invoke the emergency process.

8.2.3 Happy Path — Distressed MAP Still Online

When a MAP is in distress but still operational:

1. The CP establishes a new MAP relationship and updates DNS
2. The new MAP advertises the CP and sends `rcpidStatusRequest` to the original MAP
3. The original MAP responds with in-flight order exports
4. Other MAPs detect the directory change and update routing

If the CP appears in both directories temporarily, **conflict management** resolves it automatically using DNS verification (the CP's `_mapkey` is the tiebreaker — see §6.5).

The new MAP SHOULD advertise the distressed MAP as being in "conflict," signalling other MAPs to verify their CP associations with that MAP.

Migration can be completed in **a few hours** if the new MAP expedites CP onboarding.

8.2.4 Unhappy Path — Distressed MAP Offline

If the failing MAP cannot perform exports:

1. The CP contacts a new MAP and completes onboarding
2. The CP uses its own CRM records to identify in-flight orders
3. The new MAP issues `rcpidStatusRequest` to **other MAPs** involved in those in-flight orders
4. All new requests use new correlation IDs
5. Switch Matches without Switch Orders can be re-issued (they are not part of the export)

During the offline period:

- The CP will not appear in any directory or drop-down lists
- Messages cannot be exchanged for the affected CPs
- Any order updates must be managed with prospects and supply chains manually

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

> **NOTE**
> All MAPs can onboard new CPs within hours, so the offline period is not expected to exceed **2 working days**.

8.2.5 Per-Scenario Timing Targets

Three different commitments apply, depending on which path is in play:

SCENARIO	TARGET WINDOW	PATH
Planned MAP change (not an emergency) — normal CP transition with TTL pre-staging	**Single-digit minutes** of actual outage	§7 + the DNS TTL pre-staging routine (§3.3)
Happy-path emergency — distressed MAP still online; `rcpidStatusRequest` exports succeed	**A few hours**	§8.2.3
Unhappy-path emergency — distressed MAP offline; CRM-based reconstruction	**≤ 2 working days**	§8.2.4

These are framework targets; an individual MAP's MoU may commit to tighter numbers commercially.

8.2.6 Normal Transition vs Emergency Migration — Contrast

ATTRIBUTE	NORMAL CP TRANSITION (§7)	CP EMERGENCY MIGRATION (§8.2)
Trigger	CP-driven commercial decision	One of the §8.2.2 triggers (T1 – T6)
Notice	Planned, weeks to months	None — reactive
TTL pre-staging	Yes — 24 h+ before cut-over	Not possible (no advance notice)
New-MAP onboarding pace	Standard	Expedited — CP treated as an experienced user
Original MAP's cooperation	Required for in-flight export	Best effort; CRM fallback if offline (§8.2.4)
In-flight order recovery	Exported by old MAP via `rcpidStatusRequest`	Imported via `rcpidStatusRequest` (happy path) **or** reconstructed from CP CRM + sibling MAPs (unhappy path)
Fees / costs	Per CP – new-MAP commercial agreement	See §8.2.8 — depends on trigger
Target window	Single-digit minutes (online)	A few hours (happy) / ≤ 2 working days (unhappy)
TAG involvement	None	None unless triggered by §10 (T3 or T4)
Audit log requirement	Normal (§9)	Normal **plus** explicit emergency-event log entry (§8.2.9)

8.2.7 Emergency RACI

This RACI applies to the emergency process specifically. The in-flight-order RACI for *normal* transitions is in §7.6.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

ACTIVITY	CP	NEW MAP	OLD MAP	OTHER MAPS	TAG
Identify trigger and decide to migrate	A/R	I	I	I	I (T3/T4 only)
Select new MAP	A/R	I	I	—	—
Expedited onboarding	C	A/R	I	—	—
Generate new PKI keys; update CP DNS (`_domainkey`, `_mapkey`)	A/R	C	—	—	—
Advertise CP in new registry	I	A/R	—	I (via directory sync)	I
Issue `rcpidStatusRequest` to old MAP (happy path)	I	A/R	C (if online)	—	—
Reconstruct in-flight orders from CP CRM (unhappy path)	A/R	C	—	—	—
Issue `rcpidStatusRequest` to sibling MAPs holding the orders	I	A/R	—	C (must respond)	—
Re-issue Switch Matches lacking Switch Orders	I	C	—	A/R	—
Mark old MAP as **"in conflict"** in directory	—	A/R	—	I	I
Emergency-event audit log (§8.2.9)	C	A/R	C (if available)	C	I

R = Responsible · A = Accountable · C = Consulted · I = Informed

8.2.8 Fee Guidance

Fees for an emergency migration depend on which trigger applies:

TRIGGER	FEE TREATMENT
T1 / T2 — MAP failure or administration	New-MAP expedited-onboarding at the new MAP's **standard published rates**; the CP is not at fault. A new MAP MAY waive expedite fees as goodwill / market practice.
T3 / T4 — TAG-suspended or TAG-removed MAP	Same as T1/T2 by default. The **draft Anti-Competitive Enforcement clause** — to be published as an **Appendix to the CSF framework** once legally reviewed and signed off by TAG members (tracked as P-3 in the open-conflicts log) — would require waiver of expedite fees on the receiving side; until the Appendix is published, the T1/T2 treatment applies.
T5 — MAP unresponsive	Same as T1/T2. The CP MAY seek recovery from the unresponsive MAP under their MoU.
T6 — CP-initiated emergency	New-MAP **standard onboarding fees apply** as for a normal transition; the CP is electing the emergency path.

In all cases, the new MAP MUST publish its expedite-onboarding charge alongside its standard onboarding charge (transparency requirement, §10).

8.2.9 Audit and Retention

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Both happy- and unhappy-path emergencies generate an **emergency-event log entry** that the new MAP MUST persist. The entry SHALL contain at minimum:

- Trigger (T1 – T6 from §8.2.2) and free-text reason
- Timestamps: decision-to-migrate, DNS update, new-registry advertise, first `rcpidStatusRequest` issued, all completed
- Old MAP RCPID and new MAP RCPID
- List of `rcpidStatusRequest` correlation IDs issued and which responded
- Outcome: "happy" (all responses received) or "unhappy" (CRM reconstruction used)
- Any in-flight orders re-issued with new correlation IDs

Retention:

- **Happy path** — exported in-flight order package retained alongside the new MAP's normal CSF audit logs per the data retention policy in §9 (5 years).
- **Unhappy path** — the CP's CRM-reconstruction record forms the authoritative in-flight set for the affected interval and SHOULD be retained for the same period; the new MAP records the reconstruction inputs received from the CP.
- **All `rcpidStatusRequest` and response messages** issued or received during the emergency MUST be persistently logged per §9 audit-log requirements.

8.2.10 CP Emergency Migration Checklist

- [ ] CP contacts new MAP and agrees terms
- [ ] New MAP treats the CP as an experienced user (expedited onboarding)
- [ ] New PKI keys generated and DNS records updated (`_domainkey`, `_mapkey`)
- [ ] New MAP advertises CP in its registry
- [ ] `rcpidStatusRequest` issued to original MAP (if available) and affected sibling MAPs
- [ ] In-flight order data imported (happy) or reconstructed from CP CRM (unhappy) and reconciled
- [ ] Emergency-event audit log entry written (§8.2.9)
- [ ] Old MAP marked "in conflict" in the new MAP's directory (where applicable)
- [ ] Full operations resume

8.3 CP Merger, Acquisition, and Consolidation

CPs involved in mergers may need to consolidate brands over time. The CSF supports this through a staged process:

Stage 1 — Stop Selling

Mark consolidating brands as not accepting new Switch Matches (but still process in-flight orders and handle losses).

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Stage 2 — Asset Migration

Move customer assets from stop-sell brands to the surviving RCPID. During this period:

- The stop-sell brands remain in the directory
- The current MAP continues to handle any losses for those brands
- Care must be taken if this extends over a long period

Stage 3 — Close In-Flight Orders

Complete all remaining in-flight orders on consolidating brands. For any remaining orders:

- Complete gains
- Cancel losses where appropriate
- Migrate remaining customer assets to the surviving RCPID

Stage 4 — Remove Brands

Remove stop-sell brands from the MAP's directory. This signals to other MAPs that those CPs no longer exist. The RCPIDs are:

- Retained in time-series data for audit records
- Archived under normal data retention policies
- Never reused for different brands

8.4 CP Going into Administration

When a CP enters administration with outstanding in-flight orders:

MAP Responsibilities

1. **Hold all in-flight orders**: Switching to a CP in administration would have greater impact on end consumers than maintaining the status quo
2. **Continue processing outbound switches**: All in-flight orders transferring **away** from the CP in administration MUST continue to be processed, even if the MAP is not being paid
3. **Suspend new business**: The MAP sets the CP status to `SUSPENDED` in the directory

    - No new Switch Matches or Switch Orders will be processed
    - Existing customers are locked with their current provider under business continuity terms

1. **Await resolution**: Continue until the CP is recovered by another party or managed by an appointed insolvency team

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Industry Considerations

The response of a MAP when a CP enters administration MUST be formally defined by each individual MAP and is outside of the scope of the CSF. Refer to OTA2 guidance

8.5 Summary of Built-in Protections

SCENARIO	OTS CAPABILITY	CSF CAPABILITY
CP changes MAP	Significant manual effort	Seamless — DNS update + RCPID Status export
In-flight order portability	Not supported	Built-in via standardised JSON export
MAP failure (online)	Limited options	Automated conflict resolution + export
MAP failure (offline)	Major disruption	CP emergency migration within 2 working days
Brand consolidation	Ad-hoc	Staged process with directory controls
CP administration	Undefined	Defined responsibilities with status controls

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

9. Operations

9.1 Monitoring

Detailed monitoring specifications are to be defined by the TAG. The following principles apply:

MAPs MUST implement:

- Audit payload monitoring for all message exchanges
- Endpoint availability monitoring with automated alerting
- Anomaly detection for unusual traffic patterns

MAPs SHOULD implement:

- Transparent status dashboards accessible by connected providers
- Service status API endpoints for programmatic monitoring
- Incident and planned change indicators in the MAP registry

9.2 Malicious Behaviour Detection

In all of these scenerios is it not in the commerical interest of MAP to behave in a malicous way and all MAPs will be vetted through the onboarding process. The following scenarios MUST be monitored to detect and prevent malicious or erroneous behaviour that could happen if a MAPs was compromised. This situation is not unique to any type of architecture, hub or peer-to-peer, but within the CSF there are inherent protection mechanisms to help handle these realworld scenarios as follows:

9.2.1 Directory Hijacking

A MAP continues to advertise a CP in its directory after the CP has moved to another MAP. **Detection**: The CP will appear in two registries simultaneously, but only one DNS `_mapkey` entry will be valid. **Resolution**: DNS-based conflict resolution identifies the legitimate MAP.

9.2.2 Fraudulent CP Advertisement

A MAP advertises a CP that does not belong to it, with an incorrect root domain, attempting to send fraudulent messages. **Detection**: If all DNS entries are referentially correct, automated detection is difficult. **Resolution**: Escalation and severe penalties for the misbehaving MAP. This is monitored through the onboarding and testing process.

9.2.3 Traffic Blackholing

A MAP silently drops messages destined for one or more of its CPs. **Detection**: Sending MAPs will observe failed deliveries and timeouts. **Resolution**: Escalation via MAP contact details in the registry.

9.2.4 Denial of Service

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Due to the trusted nature of MAP-to-MAP connections, a compromised or buggy MAP could launch a denial-of-service attack against another MAP. **Detection**: Unusual traffic volume or patterns. **Resolution**: Implement standard rate limiting and the following protections:

- Reduce response frequency to make the target less appealing
- Classify the source CP as suspicious and broadcast through the directory
- Implement circuit-breaker patterns to stop processing from the offending source
- Software design practice used to prevent cascading failures in distributed systems, such as microservices. It acts like a safety switch that temporarily blocks calls to an unresponsive or failing service, allowing it time to recover while protecting the main system from wasting resources.The pattern dynamically manages traffic using three primary states:
- Closed: The system operates normally, and requests flow freely to the dependent service.
- Open: The failure rate crosses a predefined threshold. The "breaker trips," immediately blocking all requests to prevent further strain or system-wide crashes.
- Half-Open: After a set timeout, a few test requests are allowed to pass. If they succeed, the system returns to the Closed state; if they fail, it reverts to the Open state.

9.2.5 False Contact Information

A MAP advertises incorrect support contact information, redirecting support queries to malicious websites or phone numbers. **Detection**: Manual verification during onboarding and periodic audits. **Resolution**: Escalation and removal from the network.

> NOTE
> More details on detection and protection are included in the TAG's Onboarding and Testing documentation.

9.3 Standard Reporting and KPIs

9.3.1 Overview

In a distributed exchange model, there is no centralised hub for reporting. This offers advantages:

- **No expensive centralised reporting tool** to fund and maintain
- **No security risks** from a single entity tracking all industry-wide exchange patterns
- **No competition concerns** from one party having visibility of all switching traffic

MAPs have significant opportunities to provide detailed reporting statistics to their CPs and the wider community while meeting any regulatory requirements.

9.3.2 Regulatory Reporting

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

At the time of writing, there are no official Ofcom statistics that must be collected for business switching. This is recorded in the risk register (RK1). A baseline set of mandatory data point collections needs to be established, with consistent calculations published in this document. A method for sharing statistics with Ofcom may be required if they mandate submissions.

9.3.3 Industry Reporting

A standard set of industry metrics is to be defined by the TAG and will be included in subsequent versions of the CSF as part of its feature release mechanism allowing CP/MAPs to support this when they are ready.

- Switch Match success/failure rates by sending and receiving CP
- Switch Order completion rates and timelines
- In-session Switch Match failure counts (using an incremented counter in the audit envelope)
- Grouping of failures by volume per CP to identify patterns

> OTS GAP
> : The significant 1107 error ratio currently seen with OTS needs to be addressed. By enabling in-session Switch Match failure counts, the industry can distinguish between, for example, 17 attempts to achieve a successful match (which is 1 success, not 16 failures) and genuine systemic issues.

9.4 Outage Notification

The CSF outage notification mechanism is registry-native: operational status is published in the MAP Registry alongside connectivity and contact information. Because MAPs pull each other's registries at up to 5-minute intervals, any status change propagates across the network within one polling cycle — without any separate notification channel, outage calendar, or centralised hub.

The CSF considers traditional outage calendars (as used in OTS) to be outdated. A MAP suffering an incident should not need to telephone a central service desk or update a shared spreadsheet. By publishing status directly in the registry, the network self-heals: other MAPs detect the outage automatically, queue messages accordingly (see §1.3 Message Delivery Policy), and resume delivery as soon as the outage entry is cleared.

9.4.1 Design Principles

- **Registry-native**: Outage state lives in `map/outage[]` in the MAP Registry — the same pull used for CP discovery also delivers operational status.
- **Near-real-time**: At a 5-minute maximum polling interval, all connected MAPs receive updated status within 5 minutes of publication.
- **Pull-based resilience**: If a MAP fails to respond, the receiving MAP checks its cached registry for a declared outage entry. This directly informs the retry and circuit-breaker logic in §1.3.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- **Graceful degradation**: A MAP that detects a `P1` or `P2` incident on a destination SHOULD queue messages and retry rather than immediately failing the CP interaction.
- **Self-clearing**: The publishing MAP adds and removes entries. An entry with a past `end` time SHOULD be removed once service is restored. An absent or empty `outage` array means fully operational — no entry is needed to signal this state.
- **Consistent with monitoring**: The `outage[]` state SHOULD be the source of truth for the MAP's `serviceStatusURI` status page and `serviceStatusServiceURI` API — both SHOULD reflect what the registry publishes.

9.4.2 Incident Priority

Each outage entry carries an ITIL-aligned `priority` value, directly linked to the SLA fix targets in §9.9 of the Operational Excellence specification. An absent or empty `outage` array means the MAP is fully operational — no entry is needed to signal this state.

PRIORITY	§9.9 CATEGORY	CSF MEANING	RECEIVING MAP ACTION	SLA FIX TARGET
`P1`	MAP unavailable	Complete MAP outage — all services unavailable for all CPs	Queue ALL messages for ALL services; apply maximum exponential backoff per §1.3; raise an immediate P1 alert; escalate via the outage entry `contact`	Fix within 4 working hours
`P2`	CP unavailable	One or more services or CPs unavailable, or significant degradation affecting CP operations	Queue messages for affected services; apply expedited retry intervals; raise an internal alert; consult the outage entry `contact` if failures persist	Fix within 1 working day
`P3`	Service degradation	Partial or intermittent degradation — service operates but at reduced capacity or reliability	Continue delivery; apply shorter retry intervals; monitor closely; alert operations if failure rate increases	Fix within 2 working days

> **NOTE**
> An absent or empty `outage` array means the MAP is fully operational. No entry is needed to signal this state.

9.4.3 Outage Types

TYPE	WHEN TO USE
`PLANNED`	Scheduled maintenance or infrastructure change where service impact is expected. MAPs SHOULD publish planned outages at least **24 hours in advance** where possible.
`INCIDENT`	Unplanned failure or degradation. MAPs MUST publish an incident entry as soon as the incident is confirmed (target: **within 15 minutes** of detection).

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

9.4.4 JSON Structure

The `outage` array is a new field in the `map` section of the CP Registry, positioned after the `contact` array and before `connectivity`. It reuses the same `contact` sub-object for per-outage escalation details. See the full field reference in §5.4 of the CP Registry API specification.

Fully operational — empty array

```
{
  "map": {
    "name": "myMap",
    "version": "2026-06-05T08:00:00Z00000",
    "outage": []
  }
}
```

Planned maintenance window

```
{
  "map": {
    "name": "myMap",
    "version": "2026-06-06T18:00:00Z00000",
    "outage": [
      {
        "type": "PLANNED",
        "start": "2026-06-07T02:00:00Z",
        "end": "2026-06-07T04:00:00Z",
        "affectedServices": ["GPLBMessages", "OTSMessages"],
        "description": "Scheduled database maintenance. Brief interruptions of up to 30 seconds may occur. Messages will
be queued and delivered on completion.",
        "contact": {
          "type": "email",
          "value": "noc@mymapdomain.com",
          "comment": "24/7 NOC — quote this outage entry when contacting"
        }
      }
    ]
  }
}
```

Active incident (P2 — single service)

```
{
  "map": {
    "name": "myMap",
    "version": "2026-06-05T11:42:00Z00000",
    "outage": [
      {
        "priority": "P2",
        "type": "INCIDENT",
        "start": "2026-06-05T11:38:00Z",
        "affectedServices": ["GPLBMessages"],
        "description": "Letterbox API for GPLBMessages is not responding. Engineering team engaged. ETA for resolution:
30 minutes.",
        "contact": {
          "type": "phone",
          "value": "01111 999999",
          "comment": "Major incident bridge — for MAP technical contacts only"
        }
      }
    ]
  }
}
```

Active incident (P1 — complete MAP outage)

```
{
  "map": {
    "name": "myMap",
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
    "version": "2026-06-05T11:42:00Z00000",
    "outage": [
      {
        "priority": "P1",
        "type": "INCIDENT",
        "start": "2026-06-05T11:38:00Z",
        "description": "MAP infrastructure failure. All services unavailable. Engineering team engaged. ETA: 1 hour.",
        "contact": {
          "type": "phone",
          "value": "01111 999999",
          "comment": "Major incident bridge — for MAP technical contacts only"
        }
      }
    ]
  }
}
```

Multiple concurrent entries

A MAP may publish multiple entries — for example, a forthcoming planned outage alongside an active incident on a different service:

```
{
  "map": {
    "name": "myMap",
    "version": "2026-06-05T11:42:00Z00000",
    "outage": [
      {
        "priority": "P2",
        "type": "INCIDENT",
        "start": "2026-06-05T11:38:00Z",
        "affectedServices": ["GPLBMessages"],
        "description": "GPLBMessages SwitchMatchResponses unavailable. Engineering engaged. ETA: 30 minutes.",
        "contact": { "type": "phone", "value": "01111 999999", "comment": "Major incident bridge" }
      },
      {
        "type": "PLANNED",
        "start": "2026-06-07T02:00:00Z",
        "end": "2026-06-07T04:00:00Z",
        "affectedServices": ["GPLBMessages"],
        "description": "Scheduled GPLB letterbox maintenance window.",
        "contact": { "type": "email", "value": "noc@mymapdomain.com" }
      }
    ]
  }
}
```

When multiple entries are present, the **highest priority** (lowest number) for a given service determines the action taken by receiving MAPs for that service.

9.4.5 Field Reference

JSON PATH	DESCRIPTION	TYPE	REQUIRED
`map/outage[]`	Array of current incident or planned maintenance entries. An empty array or absent field means fully operational. MAPs implementing outage notification SHOULD always include this field, even as an empty array, to signal participation.	Object Array	Yes (empty if operational)
`map/outage[]/priority`	ITIL incident priority aligned to §9.9 SLA targets: `P1` (MAP unavailable — fix within 4 working hours), `P2` (CP unavailable — fix within 1 working day), `P3` (service degradation — fix within 2 working days). This is the primary severity indicator. See §9.4.2.	String (enum)	Yes for `INCIDENT`, No for `PLANNED`
`map/outage[]/type`	Outage type: `PLANNED` or `INCIDENT`. See §9.4.3.	String (enum)	Yes

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

JSON PATH	DESCRIPTION	TYPE	REQUIRED
`map/outage[]/start`	Start time of the outage or maintenance window (ISO 8601 UTC).	DateTime	Yes
`map/outage[]/end`	Expected or actual end time (ISO 8601 UTC). For active incidents where resolution time is unknown, omit until the ETA is established. MAPs MUST add or update this field as the ETA becomes known.	DateTime	Optional
`map/outage[]/affectedServices`	Routing group names (matching `map/routingGroups[]/name`) covered by this entry. If absent or empty, **all services** on this MAP are considered affected.	String Array	Optional
`map/outage[]/description`	Human-readable status narrative. SHOULD include: what is affected, current status, ETA for resolution (if known), and any available workaround. Updated as the incident evolves.	String	Yes
`map/outage[]/contact`	Escalation contact specific to this outage (e.g. NOC phone number, incident bridge). Uses the same structure as a `map/contact[]` entry.	Object	Optional
`map/outage[]/contact/type`	Contact type: `phone`, `email`, or `url`.	String (enum)	Yes (if contact present)
`map/outage[]/contact/value`	Contact value (number, address, or URL).	String	Yes (if contact present)
`map/outage[]/contact/comment`	Usage instructions — e.g. reference numbers, bridge PINs, or when to use this contact.	String	Optional

9.4.6 Receiving MAP Behaviour

When a MAP fetches another MAP's CP Registry, it MUST process the `outage` array as follows:

Step 1 — Determine effective priority per service

For each service in `map/connectivity/services[]`:

1. Collect all `outage[]` entries that reference that service in `affectedServices`, or have no `affectedServices` restriction (no restriction means all services are affected).
2. If any such entry carries `priority: "P1"`, the effective priority for that service is **P1**.
3. Otherwise, if any such entry carries `priority: "P2"`, the effective priority is **P2**.
4. Otherwise, if any such entry carries `priority: "P3"`, the effective priority is **P3**.
5. If no matching entries exist, the service is **fully operational** — proceed with normal delivery.

Step 2 — Apply delivery policy

EFFECTIVE PRIORITY	ACTION
Fully operational (no entries)	Deliver messages normally

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

EFFECTIVE PRIORITY	ACTION
P3	Continue delivery; apply shorter retry intervals; monitor closely; alert operations if failure rate increases
P2	Queue messages for affected services; apply expedited retry intervals; raise an internal alert; consult the outage entry `contact` if failures persist
P1	Queue ALL messages for ALL services; apply maximum exponential backoff per §1.3; raise an immediate P1 alert; escalate via the outage entry `contact`

Step 3 — Handle unexpected failures without a declared outage

If a MAP fails to respond and has **no** outage entry in its registry, the receiving MAP SHOULD:

1. Attempt delivery using the normal retry and backoff policy (§1.3)
2. Re-pull the source MAP's registry at the next poll cycle
3. If an outage entry subsequently appears (the MAP published one after the failure began), treat it as declared and act accordingly
4. If no entry appears after repeated failures, escalate via the MAP's `map/contact[]` entries — the absence of a declared outage when the MAP is unreachable is itself an escalation trigger

9.4.7 MAP Obligations

Planned Outages

- MAPs MUST publish a `PLANNED` entry at least **24 hours before** any maintenance with expected service impact
- The entry MUST include a realistic `end` time and a `priority` classification — typically `P3` for routine maintenance; `P2` where CP operations are expected to be significantly affected
- If the maintenance completes early, the MAP MUST remove the entry promptly
- If impact extends beyond the declared `end`, the MAP MUST update the entry immediately

Incidents

- MAPs MUST publish an `INCIDENT` entry within **15 minutes** of incident confirmation
- The entry MUST include a `priority` classification. The declared `priority` commits the publishing MAP to the corresponding SLA fix target in §9.9
- The initial entry MAY carry a brief `description` if full impact is still being assessed — it MUST be updated as the situation develops
- If the incident escalates — for example, from a single service affected (`P2`) to complete MAP unavailability (`P1`) — the MAP MUST update `priority` and `description` immediately

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

- MAPs MUST update `priority` and `description` as the incident evolves (e.g., downgrading from `P1` to `P2` on partial recovery)
- MAPs MUST remove the entry within **30 minutes** of confirmed full recovery
- Leaving stale entries causes unnecessary queuing by other MAPs — timely clearance is a network-wide obligation

9.4.8 Relationship to map.status and Monitoring

The `outage` array is distinct from the `map/status` lifecycle field:

MECHANISM	PURPOSE	VALUES
`map/status`	Administrative / lifecycle state — used for onboarding and suspension	`ACTIVE`, `TEST`, `SUSPEND`
`map/outage[]`	Operational state — real-time ITIL incident priority	`P1`, `P2`, `P3` entries (absent = fully operational)

A MAP that is `ACTIVE` in `map/status` may simultaneously carry P1/P2/P3 entries in `map/outage[]` during an incident. A MAP with `map/status: "SUSPEND"` would not normally publish outage entries — the suspension itself signals unavailability.

The `outage[]` array SHOULD be the single source of truth for the MAP's operational status pages: the `serviceStatusURI` human-readable page and the `serviceStatusServiceURI` API endpoint SHOULD derive their published status from the same underlying state, ensuring consistency between what the registry advertises and what external observers see.

9.4.9 Feature Advertising

MAPs that implement the outage notification mechanism SHOULD advertise the `outage-notification` feature in their `map/csf/features[]` array (see design proposal §6.1 of the Review Tracker):

```
"csf": {
  "versions": ["2.0"],
  "currentVersion": "2.0",
  "features": ["outage-notification", "status-api"]
}
```

A MAP that does not advertise `outage-notification` should be treated by receiving MAPs as if its `outage[]` is always empty (GREEN) — but its status signal cannot be relied upon.

9.5 Risk Register

The following risk register records risks identified across the MAP membership:

DATE	ID	DESCRIPTION	STATUS	IMPACT	LIKELIHOOD	OWNER
2025-03-20	RK1	**Regulatory and industry reporting requirements** — No official Ofcom statistics mandated	Discovery	Low	High	TBD

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

DATE	ID	DESCRIPTION	STATUS	IMPACT	LIKELIHOOD	OWNER
		for business switching. Need baseline mandatory data points with consistent open calculations. In-session Switch Match failure counting needs to be resolved.				
2025-04-09	RK2	**CP and MAP due diligence checks** — When onboarding, the sponsor is expected to complete typical credit checks, company registration, VAT registration, trading history, fraud, and ICO Data Protection registration checks. This is an industry-wide gap, not specific to CSF.	TBD	Medium	Low	TBD
2025-04-09	RK3	**Data retention policy** — Like due diligence, this is a broader industry discussion. Must address across both OTS and SforB to determine suitable retention policies and archiving/destruction methods acceptable to Data Controllers (CPs).	TBD	TBD	TBD	TBD

9.6 The Right to Block

The right to block is a **protective mechanism, not a competitive tool**. A MAP may temporarily suspend routing to or from another MAP or CP when network integrity, security, or compliance obligations require it. No central authority approves a block — the mechanism is self-administering — but each MAP's obligations under the Membership Agreement (MoU) govern when a block is legitimate, and impose strict notification and escalation duties.

9.6.1 Principles

- **Protection-first**: Blocking exists to protect the network, CPs, and end-customers — not to disadvantage competitors
- **Proportionality**: The response must match the threat. Automatic technical blocks are the lightest touch; governance escalation is the most severe
- **Transparency**: The blocking MAP MUST notify the affected party promptly and document the grounds
- **Time-limited**: All blocks MUST be reviewed at defined intervals. An indefinite block requires governance oversight
- **Self-healing**: Reinstatement SHOULD be immediate once the triggering condition is resolved

9.6.2 Grounds for Blocking

Three categories define legitimate grounds:

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Category A — Technical / Security

An immediate block is permitted without prior notification:

TRIGGER	EXAMPLE
DoS or traffic flooding	Inbound MAP-to-MAP message rate exceeds the agreed cap by more than 10×
Malformed or malicious payloads	Repeated messages failing DKIM signature validation or containing injection attempts
Credential or key compromise	A MAP's signing key is reported or confirmed compromised
Active fraud	A MAP advertises CPs it does not own (§9.2.2)

Category B — Operational / SLA

A notification attempt SHOULD be made before imposing the block:

TRIGGER	EXAMPLE
Persistent unresponsiveness	A MAP unreachable for more than 4 continuous hours with no `outage[]` entry published (§9.4)
Repeated SLA breaches	Three or more Priority 1 SLA breaches (§9.9) within a rolling 30-day window
Stale registry data	CP Registry not updated for more than 48 hours with no declared reason
Unresolved security vulnerability	A critical vulnerability unpatched beyond the agreed remediation window

Category C — Governance / MoU

Formal written notification to GPLB-SG MUST precede or accompany the block:

TRIGGER	EXAMPLE
Material MoU breach	Persistent refusal to comply with TAG-ratified obligations
Network integrity risk	Evidence that a MAP poses a systemic risk to the CSF network
Regulatory enforcement	A MAP is subject to enforcement action that affects its CSF participation

9.6.3 Blocking Procedure

Blocks operate at three tiers:

TIER	GROUNDS	DECIDED BY	MAX UNILATERAL DURATION	NOTIFICATION DEADLINE
T1 Automatic	Category A	Technical systems (circuit breaker / rate limiter)	Indefinite while threat persists; auto-clears	Notify affected MAP within 4 working hours if block persists
T2 Operational	Category B	Blocking MAP's authorised personnel	5 working days	Notify affected MAP within 2 working hours of imposing the block

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

TIER	GROUNDS	DECIDED BY	MAX UNILATERAL DURATION	NOTIFICATION DEADLINE
T3 Governance	Category C	Blocking MAP initiates; GPLB-SG oversees	Until resolved by governance	Notify affected MAP and GPLB-SG secretary within 1 working day

Tier 1 — Automatic Technical Block

Technical protections (rate limiters, circuit breakers, signature validation) activate automatically. When a block persists beyond 4 working hours, the blocking MAP's operations team MUST notify the affected MAP via its `map/contact[]` entries. The block lifts automatically when the threat condition clears, or is manually lifted after review.

Tier 2 — Operational Block

1. Before blocking, the blocking MAP SHOULD attempt to contact the affected MAP via `map/contact[]` to allow resolution
2. If no response within 2 working hours — or if immediate risk exists — the block MAY be imposed without prior contact
3. The blocking MAP MUST notify the affected MAP of: the block, the specific Category B grounds, and the conditions for reinstatement
4. The block MUST be reviewed every 5 working days. If unresolved, the blocking MAP MUST escalate to T3

Tier 3 — Governance Block

1. The blocking MAP notifies the GPLB-SG secretary and the affected MAP simultaneously, citing Category C grounds with evidence
2. GPLB-SG facilitates a resolution process within a timeframe agreed by the steering group
3. The blocking MAP may maintain the block during governance proceedings
4. Resolution may include: conditional reinstatement, a formal remediation plan, or suspension of the affected MAP's participation (`map/status: "SUSPEND"`)

9.6.4 Escalation Path

```
T1 (automatic)   →  4 working hours  →  Notify affected MAP
T2 (operational) →  5 working days   →  Escalate to GPLB-SG (T3)
T3 (governance)  →  GPLB-SG process  →  Reinstatement or suspension
```

A blocking MAP MUST NOT remain at T2 indefinitely. Escalation to T3 after 5 working days is a duty, not an option.

9.6.5 Reinstatement

Reinstatement is as lightweight as the block — no central approval is required for T1 or T2.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Tier 1: Auto-reinstated when the threat clears, or manually lifted by the blocking MAP. No formal process required.

Tier 2:

1. The affected MAP notifies the blocking MAP that the triggering condition has been resolved, with evidence
2. The blocking MAP MUST confirm or dispute within 1 working day
3. Confirmed: the block is removed immediately
4. Disputed: escalates to T3

Tier 3: Reinstatement requires GPLB-SG agreement. The affected MAP submits a reinstatement request with a remediation summary. GPLB-SG may impose conditions (e.g. monitoring period, reporting obligations) before granting full reinstatement.

9.6.6 Misuse of the Right to Block

A block imposed without legitimate grounds, or maintained beyond the point of resolution, is itself a MoU breach. An affected MAP that believes it has been blocked improperly may escalate directly to T3 — bypassing the blocking MAP — by notifying the GPLB-SG secretary with evidence. A MAP found to have blocked without grounds, or to have used blocking as a competitive measure, is subject to the same governance remedies as the blocked MAP.

9.6.7 MoU and Governance Integration

The right to block takes effect through the MAP Membership Agreement. The MoU MUST include:

- A reference to this section as the governing procedure for all blocks
- An obligation on all MAPs to maintain `map/contact[]` entries reachable within the notification windows above
- An obligation to publish and maintain `map/outage[]` (§9.4) as a precondition for challenging any Category B block raised on grounds of unresponsiveness
- An acknowledgement that T3 blocks may result in formal suspension of MAP participation

The **User Rights documentation** SHOULD reference this section to make clear to CPs that their MAP has both the right and the obligation to block other MAPs in defined circumstances — and that this mechanism exists to protect CP service continuity, not to restrict competition.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Appendix: OpenAPI Specification

Overview

This appendix contains the OpenAPI 3.0.3 specification for the CSF Letterbox API. It defines the message sending/receiving interface between MAPs, including:

- The JAM envelope structure (source, destination, routingID, auditData)
- CSF-specific HTTP headers for DKIM signing (`X-CSF-SIGNATURE`, `X-CSF-SIGNATURE-DATESTAMP`) and route tracking (`X_CSF_ROUTE`)
- OAuth 2.0 Bearer authentication
- All synchronous error responses (HTTP 4xx/5xx with 9xxx and 8xxx error codes)
- The asynchronous message delivery failure structure

This specification is compatible with the TOTSCo Hub API Specification v2.0. MAPs implementing this spec can exchange messages with both CSF MAPs and the TOTSCo Hub.

This OpenAPI specification should be reviewed alongside the Message API documentation to ensure all error codes, field descriptions, and examples are aligned. The `messageBody` is modelled as a generic object — in production, the property name should match the `routingID` (e.g., `businessSwitchMatchRequest`).

Specification

```
openapi: 3.0.3
info:
  title: CSF Letterbox API
  version: "2.0.0"
  description: |
    The Connected Services Framework (CSF) Letterbox API enables Managed Access
    Providers (MAPs) to exchange messages on behalf of their Communications
    Providers (CPs) over a secure, peer-to-peer transport layer.


    This API is compatible with the TOTSCo Hub letterbox interface (v2.0) and
    uses the industry-standard JSON Asynchronous Messaging (JAM) envelope for
    addressing, routing, and correlation.


    ## Key Characteristics


    - **Transport**: HTTPS over TLS 1.3 (mandatory)
    - **Authentication**: OAuth 2.0 Bearer Token (Client Credentials flow)
    - **Message Signing**: DKIM-based PKI via `X-CSF-SIGNATURE` header (RFC 6376)
    - **Acceptance Model**: HTTP 202 — accepted for processing, not yet completed
    - **Error Model**: Synchronous 4xx/5xx with 9xxx (industry) and 8xxx (CSF) codes


    ## Message Structure


    Every message consists of:
    1. An **envelope** containing delivery instructions (source, destination,
       routingID, auditData) — defined by the JAM Specification
    2. A **message body** containing the industry process payload (e.g., SforB
       Switch Match) — defined by the relevant process specification


    The actual message body property name MUST match the `routingID`
    (e.g., `businessSwitchMatchRequest`). This specification models it as a
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
    generic `messageBody` object for portability.


  contact:
    name: Telecom Technical Architecture Group (TAG)
  license:
    name: Open — Free for any CP or MAP to use
    url: https://github.com/nowyoyo/csf-documentation

servers:
  - url: https://{fqdn}/letterbox
    description: MAP Letterbox Endpoint
    variables:
      fqdn:
        default: api.example-map.com
        description: The MAP's Fully Qualified Domain Name (RFC 1035)

paths:
  /{version}/post:
    post:
      operationId: postMessage
      summary: Post a JAM message to the letterbox
      description: |
        Accepts a JAM-formatted JSON message and returns **202 Accepted** if
        the message passes validation and is accepted for processing.

        ### Validation Sequence

        The receiving MAP validates the message in the following order:

        1. OAuth 2.0 token validity
        2. JSON message structure
        3. Envelope attribute validation
        4. Source type and RCPID validity
        5. Source RCPID account status
        6. Destination type and RCPID validity
        7. Destination RCPID account status
        8. Source MAP authorisation via `_mapkey` DNS check (HTTP 403 + 9301 on mismatch) — see Message API §4.4.1
        9. Routing ID validity
        10. DKIM signature verification (via `X-CSF-SIGNATURE`)

        Validation stops at the first failure and the corresponding error is returned.

        ### DKIM Signing (CSF MAPs only)

        Messages between CSF MAPs MUST include `X-CSF-SIGNATURE` and
        `X-CSF-SIGNATURE-DATESTAMP` headers. Messages from HMAPs (e.g., TOTSCo)
        are not required to include these headers.

      tags:
        - Messaging
      security:
        - bearerAuth: []

      parameters:
        - name: version
          in: path
          required: true
          description: |
            Letterbox API version number. This version will only change if there
            is a substantial update in how messages are processed. Previous
            versions remain in service for backward compatibility.
          schema:
            type: string
            pattern: "^[0-9]+\\.[0-9]+$"
            example: "2.0"

        - name: X-CSF-SIGNATURE
          in: header
          required: false
          description: |
            DKIM signature for the message. Required for CSF MAP-to-MAP exchanges.
            Not required for HMAP (e.g., TOTSCo) exchanges.

            Format follows RFC 6376 with CSF-specific tags:
            `a=rsa-sha256; c=simple/simple; s={RCPID}; d={domain}; bh={bodyHash}; h=X-CSF-SIGNATURE-DATESTAMP;
b={signature}`

            See PKI & DKIM Signing documentation for full details.
          schema:
            type: string
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
          example: "a=rsa-sha256; c=simple/simple; s=809b6e65-a6e7-40f6-8b52-04dd65b6fce1; d=gplb-test.nowyoyo.net;
bh=/CWGLVBT2gdyO5cudfAEUSF43KHLmlJjU/Nr2YNqkos=; h=X-CSF-SIGNATURE-DATESTAMP; b=4J4OGZfxi..."


        - name: X-CSF-SIGNATURE-DATESTAMP
          in: header
          required: false
          description: |
            Timestamp of when the DKIM signature was created. Required whenever
            `X-CSF-SIGNATURE` is present. Embedded in the signature to protect
            against replay attacks.
          schema:
            type: string
            pattern: "^[0-9]{15}$"
            description: Format yyyyMMddHHmmssS (S = tenth of a second)
          example: "202508121340391"


        - name: X_CSF_ROUTE
          in: header
          required: false
          description: |
            Optional but recommended. Records the route the message has taken.
            Each MAP adds a new header entry (do not append to existing headers).
            Format: `{EpochMilliseconds} {MAPName}` (alphanumerics only).
          schema:
            type: string
          example: "1741947045684 TheYellowMap"


      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/LetterboxMessage'
            examples:
              sfbSwitchMatch:
                summary: SforB Switch Match Request
                description: A business Switch Match request using UUIDv4 RCPIDs
                value:
                  envelope:
                    source:
                      type: RCPID
                      identity: "d8322d80-92c8-4906-a408-f1d2daf7e03d"
                      correlationID: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
                    destination:
                      type: RCPID
                      identity: "7d87c2a7-4f64-433c-8ae1-9460f19a0a13"
                    routingID: businessSwitchMatchRequest
                    auditData:
                      - name: "channel"
                        value: "api"
                      - name: "testFlag"
                        value: "false"
                  businessSwitchMatchRequest:
                    customerPostcode: "AB12 3CD"
                    customerSurname: "SMITH"
                    businessName: "Smith Trading Ltd"


              deliveryFailure:
                summary: Message Delivery Failure Notification
                description: Asynchronous notification when a message cannot be delivered
                value:
                  envelope:
                    source:
                      type: RCPID
                      identity: "MAP-X-IDENTITY"
                    destination:
                      type: RCPID
                      identity: "d8322d80-92c8-4906-a408-f1d2daf7e03d"
                      correlationID: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
                    routingID: messageDeliveryFailure
                    auditData:
                      - name: "originalDestinationType"
                        value: "RCPID"
                      - name: "originalDestination"
                        value: "7d87c2a7-4f64-433c-8ae1-9460f19a0a13"
                      - name: "originalRoutingID"
                        value: "businessSwitchMatchRequest"
                      - name: "faultCode"
                        value: "9008"
                  messageDeliveryFailure:
                    code: "9008"
                    text: "Unable to deliver the message to the destination, timed out."
                    severity: "failure"


      responses:
        '202':
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
          description: |
            **Accepted** — The message has been accepted for processing, but
            processing has not been completed. The request might or might not
            be eventually acted upon and may be disallowed when processing occurs.


            No JSON body is returned on success.


        '400':
          description: |
            **Bad Request** — The message failed validation. This covers:
            - Invalid JSON structure
            - Missing required envelope fields
            - Invalid source/destination type or RCPID
            - Invalid routing ID
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/ValidationError'
                  - $ref: '#/components/schemas/HttpError'
              examples:
                invalidDestinationType:
                  summary: Unknown destination type (9000)
                  value:
                    errorCode: "9000"
                    errorText: "Unknown or missing destination Type."
                invalidDestinationId:
                  summary: Unknown destination RCPID (9001)
                  value:
                    errorCode: "9001"
                    errorText: "Unknown or Invalid destination."
                invalidSourceType:
                  summary: Unknown source type (9002)
                  value:
                    errorCode: "9002"
                    errorText: "Unknown or invalid source Type."
                invalidSourceId:
                  summary: Unknown source RCPID (9003)
                  value:
                    errorCode: "9003"
                    errorText: "Unknown or invalid source Id."
                invalidRoutingMapping:
                  summary: No routing ID mapped to source (9010)
                  value:
                    errorCode: "9010"
                    errorText: "No routingID is mapped with Source RCP."
                invalidRoutingId:
                  summary: Unknown routing ID (9012)
                  value:
                    errorCode: "9012"
                    errorText: "Unknown or invalid routing ID."
                schemaValidation:
                  summary: Schema validation failure
                  value:
                    code: "400"
                    message: "Bad Request"
                    description: "Schema validation failed in the Request: [Path '/directory'] Object has missing
required properties ([\"listID\"])"

        '401':
          description: |
            **Unauthorised** — Authentication failed. This covers:
            - Invalid or expired OAuth 2.0 bearer token
            - Source type and ID not permitted from originating location (9004)
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/ValidationError'
                  - $ref: '#/components/schemas/HttpError'
              examples:
                sourceNotPermitted:
                  summary: Source not permitted from this location (9004)
                  value:
                    errorCode: "9004"
                    errorText: "Source type and ID not permitted from originating location."
                invalidCredentials:
                  summary: Invalid OAuth 2.0 credentials
                  value:
                    code: "900901"
                    message: "Invalid Credentials"
                    description: "Access failure for API: /letterbox/v2.0, version: v2.0 status: (900901) - Invalid
Credentials. Make sure you have provided the correct security credentials."

        '403':
          description: |
            **Forbidden** — Authorisation succeeded but the request is not
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
            permitted. This covers:
            - Source RCPID account status is not valid (9003)
            - Destination RCPID account status is not valid (9001)
            - DKIM signature verification failed (8101 PERM_FAIL, 8102 TEMP_FAIL)
          content:
            application/json:
              schema:
                oneOf:
                  - $ref: '#/components/schemas/ValidationError'
                  - $ref: '#/components/schemas/PkiError'
              examples:
                sourceAccountInvalid:
                  summary: Source RCPID account status invalid (9003)
                  value:
                    errorCode: "9003"
                    errorText: "Source RCPD ID account status is not valid."
                destinationAccountInvalid:
                  summary: Destination RCPID account status invalid (9001)
                  value:
                    errorCode: "9001"
                    errorText: "Destination RCPD ID account status is not valid."
                pkiPermFail:
                  summary: DKIM signature permanently failed (8101)
                  value:
                    errorCode: 8101
                    errorText: "Domain gplb-test.nowyoyo.net is not valid key source for CP 809b6e65-a6e7-40f6-8b52-
04dd65b6fce1"
                pkiTempFail:
                  summary: DKIM signature temporarily failed (8102)
                  value:
                    errorCode: 8102
                    errorText: "Unable to resolve 809b6e65-a6e7-40f6-8b52-04dd65b6fce1._domainkey.gplb-test.nowyoyo.net.
TXT entry does not exist."
                pkiBodyHashMismatch:
                  summary: Body hash mismatch (8101)
                  value:
                    errorCode: 8101
                    errorText: "Body Hash 3hzlA9zA8SAVapmd5ZZnMdwaZpk/WOyQXixrqVu/WMc= is different to signature
/CWGLVBT2gdyO5cudfAEUSF43KHLmlJjU/Nr2YNqkos="
                pkiMissingTags:
                  summary: Missing mandatory signature tags (8101)
                  value:
                    errorCode: 8101
                    errorText: "Signature has missing mandatory tag(s): [b, bh]"


        '404':
          description: |
            **Not Found** — The requested resource was not found.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/HttpError'
              examples:
                notFound:
                  value:
                    code: "404"
                    message: "Runtime Error"
                    description: "No matching resource found for given API Request"


        '405':
          description: |
            **Method Not Allowed** — The HTTP method is not supported
            (e.g., GET instead of POST).
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/HttpError'
              examples:
                methodNotAllowed:
                  value:
                    code: "405"
                    message: "Runtime Error"
                    description: "Method not allowed for given API resource"


        '429':
          description: |
            **Too Many Requests** — Rate limit exceeded. The `nextAccessTime`
            field indicates when the client can retry.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ThrottledError'
              examples:
                throttled:
                  value:
                    code: "900804"
                    message: "Message throttled out"
                    description: "MAP exceeded the quota. You can access API after 2026-Apr-26 04:43:00+0000 UTC"
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
                    nextAccessTime: "2026-Apr-26 04:43:00+0000 UTC"


        '500':
          description: |
            **Internal Server Error** — An unexpected error occurred during processing.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/HttpError'
              examples:
                internalError:
                  value:
                    code: "101503"
                    message: "Runtime Error"
                    description: "Error connecting to the back end"


        '501':
          description: |
            **Not Implemented** — The HTTP method is not supported by the server.


        '502':
          description: |
            **Bad Gateway** — Invalid response from an upstream server.


        '503':
          description: |
            **Service Unavailable** — The server is temporarily unable to handle
            the request (e.g., maintenance). Clients SHOULD respect the
            `Retry-After` header if present.


        '504':
          description: |
            **Gateway Timeout** — The server did not receive a timely response
            from an upstream server.

components:
  securitySchemes:
    bearerAuth:
      type: http
      scheme: bearer
      bearerFormat: JWT
      description: |
        OAuth 2.0 Bearer Token obtained via the Client Credentials flow.
        Request a token from the target MAP's OAuth token endpoint using
        the `client_id` and `client_secret` exchanged during onboarding.

  schemas:

    # ──────────────────────────────────────────────
    # Request Body
    # ──────────────────────────────────────────────

    LetterboxMessage:
      type: object
      required:
        - envelope
      properties:
        envelope:
          $ref: '#/components/schemas/Envelope'
      additionalProperties:
        type: object
        description: |
          The message body. The property name MUST match the `routingID`
          in the envelope (e.g., `businessSwitchMatchRequest`,
          `messageDeliveryFailure`). Modelled here as `additionalProperties`
          because the key is dynamic.
      description: |
        A JAM-formatted message consisting of an envelope and a
        process-specific message body.

    Envelope:
      type: object
      required:
        - source
        - destination
        - routingID
      properties:
        source:
          $ref: '#/components/schemas/EndpointRef'
          description: |
            The originator of the message and return address for any
            response. The `correlationID` MUST always be provided in source.
        destination:
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
        $ref: '#/components/schemas/EndpointRef'
        description: |
          The intended recipient. The `correlationID` is only present in
          response messages — it contains the original sender's correlationID.
      routingID:
        type: string
        description: |
          Determines the message type and routing. Must be a value supported
          by the recipient MAP's routing groups. Valid values include
          `businessSwitchMatchRequest`, `businessSwitchOrderRequest`,
          `residentialSwitchMatchRequest`, `messageDeliveryFailure`,
          and `notification`.
        example: businessSwitchMatchRequest
      auditData:
        type: array
        description: |
          Name-value pairs for regulatory reporting, diagnostics, and
          industry auditing. Each industry process defines its own audit
          requirements. The `testFlag` audit field MUST be set to `"true"`
          for all test traffic.
        items:
          $ref: '#/components/schemas/AuditDatum'


  EndpointRef:
    type: object
    required:
      - type
      - identity
    properties:
      type:
        type: string
        description: |
          The directory list type. Currently only `RCPID` is supported.
        enum:
          - RCPID
        example: RCPID
      identity:
        type: string
        description: |
          The RCPID of the sender (source) or recipient (destination).
          For CSF, this is a UUIDv4 (36 characters). For TOTSCo
          compatibility, 4-character OTS codes are also accepted.
        example: "d8322d80-92c8-4906-a408-f1d2daf7e03d"
      correlationID:
        type: string
        description: |
          In source: MUST always be provided. Should be sufficiently unique
          to allow correlation of responses over a reasonable period. UUIDs
          are recommended.


          In destination: only present in response messages, containing the
          original sender's correlationID.
        example: "a1b2c3d4-e5f6-7890-abcd-ef1234567890"


  AuditDatum:
    type: object
    required:
      - name
      - value
    properties:
      name:
        type: string
        description: The audit field name
        example: "channel"
      value:
        type: string
        description: The audit field value
        example: "api"


  # ──────────────────────────────────────────────
  # Asynchronous Delivery Failure Body
  # ──────────────────────────────────────────────


  MessageDeliveryFailure:
    type: object
    required:
      - code
      - text
      - severity
    properties:
      code:
        type: string
        description: |
          Numeric fault code for remedial action. Uses 9xxx series codes.
        example: "9008"
      text:
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
        type: string
        description: Human-readable description of the fault
        example: "Unable to deliver the message to the destination, timed out."
      severity:
        type: string
        description: |
          Nature of the notification.
        enum:
          - information
          - warning
          - failure
        example: failure
    description: |
      Sent asynchronously by a MAP when it cannot deliver a message to the
      intended recipient. The envelope source is the MAP that failed to
      deliver, and the envelope destination contains the original sender's
      correlationID.


      ### Delivery Failure Codes (9xxx)


      | Code | Description | Severity |
      |------|-------------|----------|
      | 9005 | No valid route to destination | failure |
      | 9006 | Rejected — invalid message format (HTTP 400) | failure |
      | 9007 | Recipient rejected message (HTTP 404) | failure |
      | 9008 | Timed out after retries (HTTP 5xx or timeout) | failure |
      | 9013 | Invalid API key | failure |
      | 9014 | API key expired | failure |
      | 9015 | Digital certificate invalid | failure |
      | 9016 | Digital certificate expired | failure |


  # ─────────────────────────────────────────────
  # Error Response Bodies
  # ─────────────────────────────────────────────


  ValidationError:
    type: object
    required:
      - errorCode
      - errorText
    properties:
      errorCode:
        type: integer
        description: |
          Numeric error code from the 9xxx (industry) or 8xxx (CSF) series.
        example: "9002"
      errorText:
        type: string
        description: |
          Human-readable description of the error with suggested
          remedial action.
        example: "Unknown or invalid source Type."
    description: |
      Returned for validation errors detected during envelope processing.
      Uses 9xxx codes for industry-standard validations and 8xxx codes
      for CSF PKI errors.


      ### Validation Error Codes (9xxx)


      | Code | HTTP | Description |
      |------|------|-------------|
      | 9000 | 400 | Unknown or missing destination type |
      | 9001 | 400 | Unknown or invalid destination RCPID |
      | 9001 | 403 | Destination RCPID account status invalid |
      | 9002 | 400 | Unknown or invalid source type |
      | 9003 | 400 | Unknown or invalid source RCPID |
      | 9003 | 403 | Source RCPID account status invalid |
      | 9004 | 401 | Source not permitted from originating location |
      | 9010 | 400 | No routing ID mapped to source RCP |
      | 9012 | 400 | Unknown or invalid routing ID |


      ### PKI Error Codes (8xxx)


      | Code | HTTP | Description | Retry? |
      |------|------|-------------|--------|
      | 8101 | 403 | PERM_FAIL — signature verification permanently failed | No |
      | 8102 | 403 | TEMP_FAIL — signature verification temporarily failed | Yes |


  PkiError:
    type: object
    required:
      - errorCode
      - errorText
    properties:
```

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

```
        errorCode:
          type: integer
          enum:
            - 8101
            - 8102
          description: |
            8101 = PERM_FAIL (do not retry)
            8102 = TEMP_FAIL (retry with backoff)
        errorText:
          type: string
          description: Detailed description of the PKI failure
      description: |
        Returned with HTTP 403 when DKIM signature verification fails.
        Only used for CSF MAP-to-MAP exchanges (not sent to TOTSCo).

    HttpError:
      type: object
      properties:
        code:
          type: string
          description: HTTP status code or internal process code
          example: "400"
        message:
          type: string
          description: Short error classification
          example: "Bad Request"
        description:
          type: string
          description: Detailed error description with remedial guidance
          example: "Schema validation failed in the Request"
      description: |
        Generic HTTP error response. Used for schema validation failures,
        authentication errors, and server-side errors.

    ThrottledError:
      type: object
      properties:
        code:
          type: string
          description: Internal process code (not an HTTP error code)
          example: "900804"
        message:
          type: string
          example: "Message throttled out"
        description:
          type: string
          example: "MAP exceeded the quota."
        nextAccessTime:
          type: string
          description: |
            When the client can retry. Format varies by implementation.
          example: "2026-Apr-26 04:43:00+0000 UTC"
      description: |
        Returned with HTTP 429 when the receiving MAP's rate limit is exceeded.
```

Usage Notes

Content-Type

All requests MUST use `Content-Type: application/json` with charset UTF-8.

TLS Requirement

All connections MUST use TLS 1.3. Earlier TLS versions MUST be rejected.

DKIM Headers

For CSF MAP-to-MAP exchanges, the `X-CSF-SIGNATURE` and `X-CSF-SIGNATURE-DATESTAMP` headers are **mandatory**. For messages from HMAPs (e.g., TOTSCo), these headers are **optional**.

When a MAP forwards a message (e.g., TOTSCo passing a CSF-signed message to its CPs), the `X-CSF-SIGNATURE` header SHOULD be left intact for end-to-end verification.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.

Route Tracking

The `X_CSF_ROUTE` header is optional but recommended. Each MAP that handles the message adds a **new** header (not appended to existing headers). The final recipient can read all `X_CSF_ROUTE` headers to reconstruct the message route.

Dynamic Message Body Key

The JAM specification allows the message body property name to match the `routingID`. For example, a message with `routingID: "businessSwitchMatchRequest"` would have its body under:

```
{
  "envelope": { ... },
  "businessSwitchMatchRequest": { ... }
}
```

The OpenAPI specification models this using `additionalProperties` on the root `LetterboxMessage` object, since the key name is dynamic.

Error Response Formats

The CSF uses two distinct error response formats, each governed by an industry standard that the framework cannot modify. This is a statement of fact, not a review item:

1. **`ValidationError`** (`errorCode` / `errorText`) — for 9xxx and 8xxx validation and PKI errors. This format is defined by the industry-standard JSON Asynchronous Messaging (JAM) Specification and is used consistently across CSF and TOTSCo implementations. Changing it would break compatibility with the wider switching ecosystem.
2. **`HttpError`** (`code` / `message` / `description`) — for HTTP-level and schema errors. This format follows established IETF RFC standards for HTTP error responses and cannot be modified without breaking standards compliance.

Implementers MUST support receiving both formats. The dual-format design reflects the layered nature of the CSF (industry application protocol over HTTP transport) — it is not transitional and is not under review.

© 2025–2026 Telecom Technical Architecture Group (TAG). All rights reserved.