Gallagher Command Centre

Modular Platform - App Host and Message Bus

Technical Information Paper

Copyright © Gallagher Group Limited

Disclaimer

This document gives certain information about products and/or services provided by Gallagher Group Limited or its related companies (referred to as "Gallagher Group").

The information is indicative only and is subject to change without notice meaning it may be out of date at any given time. Although every commercially reasonable effort has been taken to ensure the quality and accuracy of the information, Gallagher Group makes no representation as to its accuracy or completeness and it should not be relied on as such. To the extent permitted by law, all express or implied, or other representations or warranties in relation to the information are expressly excluded.

Neither Gallagher Group nor any of its directors, employees or other representatives shall be responsible for any loss that you may incur, either directly or indirectly, arising from any use or decisions based on the information provided.

Except where stated otherwise, the information is subject to copyright owned by Gallagher Group and you may not sell it without permission. Gallagher Group is the owner of all trademarks reproduced in this information. All trademarks which are not the property of Gallagher Group, are acknowledged.

Copyright © Gallagher Group Ltd 2026. All rights reserved.

Contents

1 Background 4
1.1 Objective 4
1.2 What the App Host and Message Bus are 4
2 Architecture 5
2.1 Overview 5
2.2 The App Host 6
2.3 The Message Bus 6
2.4 Relationship to Command Centre, and how it will evolve 6
3 Deployment and Hosting 7
4 Scale and Resilience 8
5 Modules: Hosting and Client UI Integration 8
6 Authentication and Authorisation 8
7 Data Transmission and Communications Security 9
7.1 Command Centre Client to Application Host 9
7.2 Message Bus communications 9
7.3 Certificates, keys and TLS configuration 9
8 Data Storage and Retention 10
9 Security Controls 11
10 Logging and Diagnostics 11
11 Standards and Conventions 11
12 Security and Penetration Testing 12
13 Status of this Document 12

Copyright © Gallagher Group Limited

1 Background

Gallagher Command Centre is a mature security platform that customers deploy and host in their own environment, and that has been developed and extended over many years. Historically it has been delivered as a single, large application on a long release cadence. To support more frequent and more flexible delivery of functionality, Gallagher is progressively introducing a modular platform, in which discrete capabilities are delivered as independent modules that run alongside the existing Command Centre Server.

Two components provide the foundation of this modular platform: the App Host and the Message Bus. Together they provide a common hosting and communication layer, so that each module does not have to solve hosting, routing, integration, security and messaging concerns for itself. Both are installed as part of Command Centre, run on the same customer-hosted infrastructure, operate entirely in the background, and do not require operator interaction.

This paper describes the current generation of the App Host and Message Bus — the architecture in which modules are hosted in-process within a single App Host service that also provides the gateway. This generation spans the Command Centre releases noted in Section 3 (Deployment and Hosting), and remains current until the next-generation architecture (a separated gateway and independently hosted modules, outlined in Section 4, Scale and Resilience) is released. It reflects the current state of an actively evolving platform; the direction of travel is summarised in Section 2.4 (Relationship to Command Centre), Section 4 (Scale and Resilience) and Section 13 (Status of This Document).

1.1 Objective

The objective of this paper is to give customers and channel partners a clear, factual understanding of the App Host and Message Bus from a standards, architecture, security and general-use standpoint.

These components are visible at a customer level — they appear as services running alongside Command Centre on the server. This paper explains their role, their deployment footprint, and how they fit within the security and network boundary of a Command Centre installation. The authoritative reference for the cryptography these components use is the Command Centre Cryptography and Encryption Technical Information Paper; this paper summarises that information and cross-references it.

1.2 What the App Host and Message Bus are

- The **App Host** is a Windows service that provides a common hosting platform for Command Centre modules — a key building block in Command Centre's transition toward a modular, distributed architecture. It loads modules, routes requests to them, resolves the permissions they need, and makes the administration UIs that modules provide discoverable to the Command Centre Client. The goal is that module developers can build functionality without having to build their own hosting, integration and security plumbing, and that customers are not exposed to a proliferation of separate services to manage.
- The **Message Bus** is a messaging service that modules and other platform services use to communicate in a decoupled way. It is designed to support high-volume communication while keeping services independent of one another, improving system resilience and performance. A component publishes a message on a named *subject*, and the bus delivers it to whichever components have subscribed to that subject; components do not address messages to one another by name.

Modules are the units of functionality that run on this platform. Today a small number of modules are in production; over time, capabilities currently delivered within the Command Centre Server are expected to be delivered as modules (see Section 2.4, Relationship to Command Centre).

Copyright © Gallagher Group Limited

2 Architecture

2.1 Overview

The App Host and Message Bus run within the customer's existing, **customer-hosted** Command Centre environment, on the Command Centre Server. They do not introduce a separate hosting location and do not require any Gallagher-hosted cloud service to operate.

In the current generation of the platform:

- The **App Host** is a single Windows service that hosts the platform's modules **in-process** and also performs the **gateway / reverse-proxy** role (routing and request handling) within that same service.
- The **Message Bus** is a separate Windows service that carries messages between the Command Centre Service, the App Host and each App Host module.
- The existing **Command Centre Server** continues to operate as it does today. Modules integrate with it rather than modifying it, and the Command Centre Client retains its existing connection to the Command Centre Server.

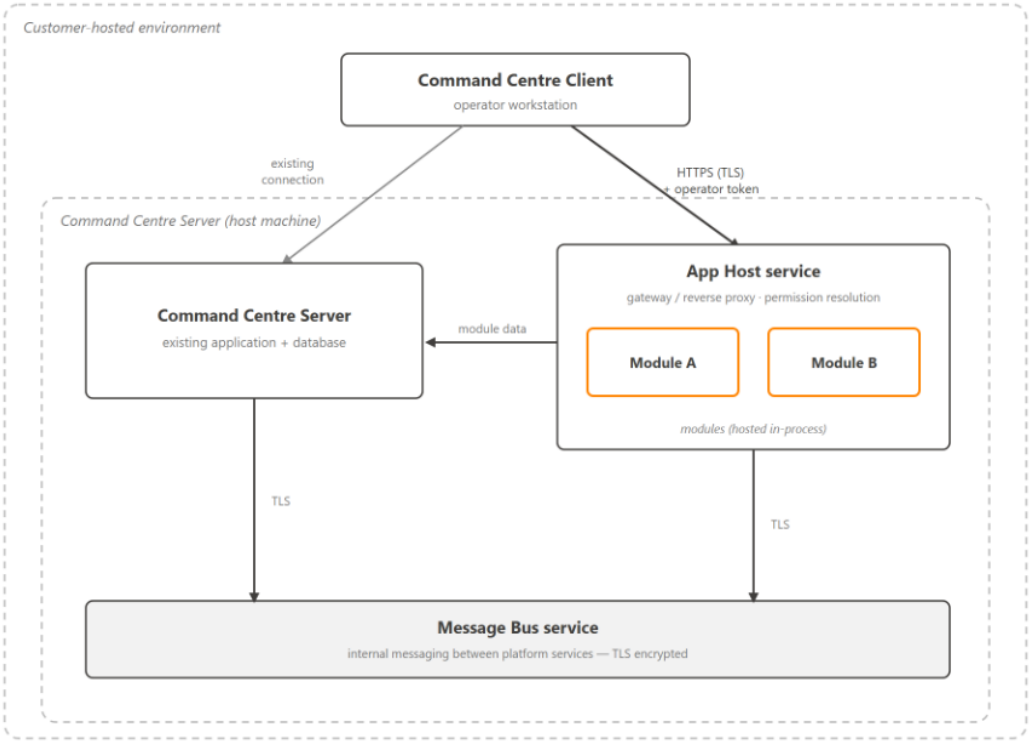

Figure 1:
The App Host (gateway and modules combined in one service) and the Message Bus, on the Command Centre Server.

The architecture is designed to evolve so that these components can be separated and run as multiple instances — enabling load balancing and high availability — as described in Section 4 (Scale and Resilience).

Copyright © Gallagher Group Limited

2.2 The App Host

The App Host is a single Windows service that hosts modules and routes requests to them. Its responsibilities are:

- **Module hosting.** On startup, the App Host loads the installed modules from a directory within its installation folder, each kept isolated from the others, and starts them. In this generation, modules are hosted within the App Host process itself and run with the privileges of the App Host service account.
- **Request routing.** The App Host provides an HTTPS endpoint that the Command Centre Client uses to make requests to modules, and routes each incoming request to the appropriate module over a private, local inter-process channel. (This endpoint is in addition to the Command Centre Client's existing connection to the Command Centre Server, which it retains.)
- **Permission resolution.** For each incoming request that carries an authenticated operator session, the App Host resolves the operator's relevant permissions and makes them available to the module, so each module does not have to perform that resolution itself (see Section 6, Authentication and Authorisation).
- **Making module UIs discoverable.** The App Host allows modules to contribute administration UI to the Command Centre Client without the client having to be changed for each new module. It does this by making the available module UIs *discoverable* (see Section 5, Modules); the App Host itself does not render those UIs.

Combining these responsibilities into a single managed service is deliberate: it means a site runs one platform service rather than a separate Windows service for every module, reducing the management overhead for installers and customers.

Because modules currently run within the App Host process, there is no independent restart of an individual module in this generation: if a module needs to be restarted, the App Host service is restarted, which restarts the modules it hosts. Independent supervision and restart of individual modules is part of the platform's future direction (see Section 4, Scale and Resilience, and Section 13, Status of This Document).

2.3 The Message Bus

The Message Bus is a messaging service that modules and other platform services use to communicate without being directly coupled to one another. It is built on NATS, an open-source messaging system in which services exchange messages by publishing and subscribing to named subjects rather than connecting to each other directly (https://nats.io). It supports the common messaging patterns — publish/subscribe (announcements) and request/reply (a query and its responses) — and is designed for high-volume communication that keeps services independent, improving resilience and performance.

The Message Bus runs as a Windows service on the Command Centre Server and uses the NATS protocol on a local TCP port (8444 by default). It is co-located with the server to keep installation straightforward. The Message Bus is an internal communication mechanism of the platform — it is not an externally exposed interface, and customers and partners do not interact with it directly. Communication with the Message Bus uses mutually authenticated TLS (see Section 7.2, Message Bus communications).

2.4 Relationship to Command Centre, and how it will evolve

The modular platform is being introduced **alongside** the existing Command Centre Server, not as a replacement that requires the existing system to be re-engineered. It lets sites run next-generation Gallagher services alongside the existing application, helping them modernise their

Copyright © Gallagher Group Limited

Command Centre deployment without disruption. The existing server continues to operate as it does today, and modules integrate with it.

Over time, functionality that is currently delivered within the Command Centre Server is expected to move onto this platform as modules. This is a deliberately **gradual migration**: existing functionality continues to operate unchanged while new capability is delivered incrementally, so that customers are not exposed to disruptive change. The pace and scope of that migration will be guided by Gallagher's release planning over time.

3 Deployment and Hosting

The App Host and Message Bus are deployed on the same Microsoft Windows Server as Command Centre, and are installed automatically as part of the Command Centre server installation package — no separate installer is required. They are **customer-hosted**: they run within the customer's own Command Centre environment — whether that is on the customer's premises, in their data centre, or in their own cloud infrastructure — rather than being hosted by Gallagher. The Message Bus is installed from Command Centre 9.40 onwards; the App Host is installed from Command Centre 9.50 onwards (and is compatible with 9.40 for out-of-band installation where required). Each is a standard Windows service — the "Gallagher Command Centre AppHost Service" and the "Gallagher Command Centre Message Bus".

Key deployment characteristics:

- **Co-located with Command Centre.** At present, both services run on the Command Centre Server itself, and must be running for Command Centre to operate correctly.
- **Low-privilege service accounts.** Both services are designed to run under a low-privilege Windows account — a standard user account granted the "Log on as a service" right (the only additional right required beyond standard user rights), with no elevated or administrative privileges at runtime. By default they run under the Local Service account, or under the dedicated account selected during Command Centre installation. During installation, Command Centre grants the account the specific permissions each service needs (such as its Command Centre database login, file and registry permissions, and access to the private key of the certificate it uses to authenticate to the Message Bus). Service-account guidance is covered in the Command Centre Hardening Guide.
- **No dependency on external connectivity.** The App Host and Message Bus do not, of themselves, require or establish any connection to a service outside the customer's own environment. They operate entirely within the customer's environment.
- **Network ports.** The App Host exposes an HTTPS endpoint on TCP port **8910** that Command Centre Client workstations (and other applications acting on behalf of an operator) connect to over the customer's network. The Message Bus listens locally on TCP port **8444**. Introducing the App Host requires port 8910 to be reachable by Command Centre Client workstations; the Command Centre installer creates an inbound firewall rule for the App Host, and firewall scoping may need to be reviewed. The Command Centre Hardening Guide, Release Notes and installation documentation cover the ports and firewall configuration.
- **Same operational boundary as Command Centre.** Because the components run within the customer's existing environment, they sit within the network, physical and operational security boundary already established for that environment.
- **Reduced management overhead.** A single App Host service manages the modules on a machine, so installers and customers are not faced with a separate service to install and monitor for every module.

Copyright © Gallagher Group Limited

4 Scale and Resilience

Current generation. At present, the App Host runs as a single service on the Command Centre Server, with the gateway and modules combined in that one service, and the Message Bus co-located on the same server. Scale and availability are therefore those of that single server: the platform is designed for correct, straightforward operation within a customer's existing Command Centre deployment rather than for horizontal scale-out at this stage.

Future direction. A central goal of this architecture is to support horizontal scalability and high availability as it matures. The intended approach is to run **multiple instances of a module across multiple machines**, with the **gateway running as a separate component**, external to the App Host processes, able to route and load-balance across those instances. This is expected to be the primary mechanism for scalability and resilience, and to give greater flexibility around upgrades (for example, updating one instance at a time). Supporting this, the platform is being designed so that the gateway, App Host and Message Bus can be separated and run as multiple instances rather than as the single co-located set described above.

These capabilities are **not** part of the current generation. They are described here to explain the direction of the architecture; nothing in this section is a commitment to deliver a particular capability on a particular timeframe. This document will be updated as these capabilities are delivered.

5 Modules: Hosting and Client UI Integration

A module is a unit of functionality that runs on the platform. It typically provides a server-side component (hosting a web service and any supporting APIs) and may also provide administration UI that is shown inside the Command Centre Client.

Installation. Modules are typically packaged as part of Command Centre and installed by the integrated installer, so that a customer receives current, compatible modules with their Command Centre installation. Modules can also be installed independently of the Command Centre installer — this allows a module to be released out of band with the main Command Centre release cadence where there is a reason to do so. In either case, installing a module makes it available to the App Host, which loads and runs it.

Client UI integration. So that new modules can contribute administration UI to the Command Centre Client without the client having to be modified for each one, the App Host makes the available module UIs *discoverable*. When an operator logs in, the Command Centre Client asks the App Host which module administration UIs are available, and the App Host returns a consolidated list gathered from the installed modules. To display one of those UIs, the Command Centre Client then retrieves it from the relevant module through the App Host gateway. The discovery request carries the operator's context and selected language, so modules can return appropriately filtered and localised entries. Currently, this supports module administration views.

6 Authentication and Authorisation

Requests made to modules through the App Host are authenticated and authorised on a per-request basis.

- **Operator authentication.** When the Command Centre Client makes a request to the App Host, it includes a short-lived security token (a JSON Web Token, or JWT) that identifies the current operator session. The token is issued and signed by the Command Centre Server using ECDSA P-384 with SHA-384, and asserts the operator's identity together with their privileges and divisions. Its default lifetime is five minutes, configurable to a maximum of sixty minutes. The App Host validates the token before processing the request.

Copyright © Gallagher Group Limited

- **Signing-key validation and rotation.** The Command Centre Server publishes the public certificate used to verify these tokens to the Message Bus, so the App Host can validate them. The App Host caches the certificate; because it is distributed over the Message Bus, the signing certificate can be changed (rotated) over time. If the signing certificate is changed, previously issued tokens no longer verify.
- **Per-route protection.** Each route that a module exposes declares how it is protected. The App Host enforces this: *authenticated* routes require a valid operator token before the request is forwarded; *client-certificate* routes require the caller to present a TLS client certificate; and *anonymous* routes may be used where no authentication is appropriate. On a client-certificate route, a request without a certificate is rejected with HTTP 401; otherwise the App Host forwards the presented certificate to the module, which is responsible for validating it and deciding whether the caller is authorised.
- **Authorisation.** For an authenticated operator request, the operator's identity, privileges and divisions (asserted by the token and resolved by the App Host) are made available to the module, which grants access on that basis.

7 Data Transmission and Communications Security

All communication involving the App Host and Message Bus takes place **within the customer's own Command Centre environment**. These components do not, of themselves, transmit data outside that environment. The authoritative catalogue of the cryptography used is the Command Centre Cryptography and Encryption Technical Information Paper, to which the App Host and Message Bus have been added; this section summarises the key points.

7.1 Command Centre Client to Application Host

Communication between the Command Centre Client and the App Host uses HTTPS on TCP port 8910, with the operating system's configured TLS settings (TLS 1.3 or higher recommended). The App Host presents a TLS server certificate; its thumbprint is supplied to the client by the Command Centre Server when the client establishes its session, and the client verifies the App Host against that thumbprint. Command Centre Clients do not present a certificate on this connection — the operator is authenticated by the signed token described in Section 6, Authentication and Authorisation.

7.2 Message Bus communications

The Command Centre Service, the App Host and App Host modules can communicate with one another through the Message Bus, which uses the NATS protocol on TCP port 8444. (The Message Bus is one of the available mechanisms — modules are also reached over HTTP through the App Host gateway, as described in Section 2.2.) Connections use mutually authenticated TLS: the Message Bus and each connecting participant hold their own self-signed certificate, created during installation, and each participant verifies the Message Bus against its exported certificate thumbprint rather than against a certificate authority.

Authorisation is separate from transport security. Each service may publish and subscribe only to the subjects registered for it during installation. These permissions are carried in a token signed by Gallagher, which the installer wraps in a second token signed by a certificate unique to that installation before registering it with the Message Bus. The Message Bus accepts only the certificates in its pinned certificate file.

7.3 Certificates, keys and TLS configuration

- **Keys and certificates.** The certificates used by these components are ECDSA on the NIST P-384 elliptic curve with SHA-384, and keys are generated at the time of installation. The App

Copyright © Gallagher Group Limited

Host gateway certificate and the Message Bus certificates are self-signed and created by the installer; the App Host gateway certificate includes subject alternative names covering the server's host name, fully qualified domain name, localhost and IP addresses, and is valid for ten years.

- **Customer-supplied certificates.** The certificate used to sign operator tokens can be a site's own: a site may use the internal default certificate, supply its own certificate to be stored in the Command Centre database, or use a certificate held in the Windows certificate store on the server.
- **TLS protocol and cipher suites.** The TLS protocol versions and cipher suites in use are governed by the host operating system's configuration (for example, Windows Schannel). TLS 1.3 is negotiated where the operating system supports and enables it; otherwise TLS 1.2 is used. These components rely on the operating system's TLS settings rather than enforcing their own.
- **FIPS 140.** The Message Bus is built to operate in a manner consistent with FIPS 140-3.

The channels summarise as follows (see the Cryptography and Encryption TIP for the full table):

Channel	Key Strength	Encryption	Key storage
Command Centre Client to Application Host	ECC P-384	AES-256 (TLS 1.2 / 1.3 depending on OS configuration)	Windows Certificate Store
Command Centre services to Message Bus	ECC P-384	AES-256 (TLS 1.2 / 1.3 depending on OS configuration)	Windows Certificate Store

The App Host and Message Bus do not change how Command Centre communicates with external systems. Any communication between Command Centre and external services or integrations continues to be governed by the existing Command Centre mechanisms.

8 Data Storage and Retention

The App Host and Message Bus are platform components; the business data of the system continues to belong to Command Centre and its modules.

- **Module data.** A module stores its data in its own private schema within the Command Centre database environment. Because this data sits within the Command Centre database, it is covered by Command Centre's standard backup and disaster-recovery processes. When a database is restored, modules update their own schema as required on startup.
- **Message Bus content.** To support reliable delivery, the Message Bus retains messages on disk (using NATS JetStream) under `%ProgramData%\Gallagher\Message Bus\Command Centre`. These retained messages are stored unencrypted at the application level and should be protected by full-disk encryption, as recommended in the Command Centre Hardening Guide. The Message Bus is a transport mechanism, not a long-term system of record. Under normal operation the volume of retained messages is small, because messages are removed once all subscribers have received them. Storage use can grow in some circumstances — for example if a subscribing service is stopped or is unable to process messages — so free space on the server's system volume should be monitored using the site's standard server monitoring.
- **No customer data leaves the environment.** Because these components operate entirely within the customer's environment, they do not transmit customer data outside it.

Operational and diagnostic information (such as log files) is generated to support running and supporting the platform, and is held within the customer's environment (see Section 10, Logging and Diagnostics).

Copyright © Gallagher Group Limited

9 Security Controls

The App Host and Message Bus are developed and delivered as part of Command Centre and are subject to the same secure-development and security-assurance processes as the rest of the product.

- **Within the existing trust boundary.** The components run inside the customer's existing Command Centre environment and sit within its established security controls. They do not introduce connectivity to a service outside the customer's environment.
- **Low-privilege operation.** Both services run under low-privilege Windows accounts with no elevated privileges at runtime (Section 3).
- **Authenticated access.** Requests through the App Host are protected according to each route's declared protection level, with operator requests authenticated using a short-lived, Command Centre-signed token (Section 6).
- **Authenticated, least-privilege messaging.** Services authenticate to the Message Bus with their own certificate under mutual TLS, and are restricted to the specific subjects they were authorised to use at install time (Section 7.2).
- **Module isolation.** Modules are loaded in isolation from one another within the App Host. Stronger, process-level isolation between modules is part of the platform's future direction (see Section 4, Scale and Resilience, and Section 13, Status of This Document).
- **Maintained with Command Centre.** Security updates to these components are delivered through the normal Command Centre release and update process. Where necessary, an update may be provided out of band.

Hardening guidance for these components — including service accounts, certificate key protection, firewall scoping and full-disk encryption of retained data — is provided in the Command Centre Hardening Guide.

10 Logging and Diagnostics

The platform produces operational information to support running and supporting a site:

- **App Host.** The App Host writes its own log files to `%ProgramData%\Gallagher\AppHost Service`. It may also emit telemetry (such as traces, metrics and structured log events) to the Command Centre Diagnostics Service.
- **Message Bus.** The Message Bus writes its own log files (generated by the embedded NATS runtime) to `%ProgramData%\Gallagher\Message Bus\Command Centre`. These are not emitted as telemetry to the Diagnostics Service.

Log files from both components are collected by Gallagher's support tooling (such as the Support Log Tool) when diagnostic information is gathered for Technical Support. Log locations are configurable; the Command Centre Release Notes document the defaults and the relevant settings.

11 Standards and Conventions

The App Host and Message Bus are built on established, widely understood technologies and patterns rather than bespoke mechanisms:

- **Standard hosting and transport.** The components are delivered as standard Windows services, and communication uses HTTPS / TLS. TLS protocol versions and cipher suites follow the host operating system's configuration (see Section 7.3).

Copyright © Gallagher Group Limited

- **Token-based authentication.** Operator authentication uses signed JSON Web Tokens (JWTs), a widely adopted standard, with support for signing-certificate rotation.
- **Open, subject-based messaging.** The Message Bus is built on NATS, an open-source messaging system. Services communicate by publishing and subscribing to named subjects rather than connecting to each other directly — a standard approach for building modular, independently maintainable systems. The messaging technology is accessed through a common interface so that modules are insulated from its implementation details.
- **Elliptic-curve cryptography.** Certificates and keys use ECDSA on the NIST P-384 curve with SHA-384 (see Section 7.3).
- **Recognised cryptographic standards.** The Message Bus is built to operate in a manner consistent with FIPS 140-3.

Adopting recognised patterns and conventions improves interoperability, supportability and the long-term maintainability of the platform.

12 Security and Penetration Testing

The App Host and Message Bus are developed within Gallagher's established security assurance processes for Command Centre.

Gallagher employs internal security and penetration testing staff who hold a number of security certifications and who play a key role in the development of the product, providing expertise, security reviews and internal penetration testing. Gallagher also engages an external specialist security company, CyberCX (previously Insomnia Security), to conduct security reviews and penetration testing. A summary of CyberCX's assessment is available to channel partners in the Gallagher Group Security Assessment Statement, at: https://media.gallagher.com/asset/3753e65f-0fb0-44a1-a21f-d3a098e11054/Gallagher-Group-Security-Assessment-Statement-Letter-20260604.pdf

13 Status of this Document

The modular platform described in this paper is **actively evolving**. The App Host and Message Bus are the foundational building blocks of Command Centre's modular platform and are in production today.

Their capabilities will continue to expand as functionality is gradually migrated onto the platform. Anticipated directions include process-level isolation of modules, independent supervision and restart of individual modules, and horizontal scalability and high availability through running multiple instances of a module across multiple machines with a separate gateway (see Section 4, Scale and Resilience), along with additional telemetry.

This edition describes the platform as it stands at the date shown on the cover. It does not describe future capability, and nothing in this paper should be read as a commitment to deliver any particular capability, on any particular timeframe. Customers and partners should refer to the latest edition of this document, and to Gallagher's release communications, for the current state.

Copyright © Gallagher Group Limited