



FlowMart - Governance Report

Governance review, score and recommendations on your architecture

May 26, 2025

Report Overview

This Report, generated from FlowMart's EventCatalog on **May 26, 2025**, provides a comprehensive snapshot of your organization's documented architecture. Its primary objective is to offer key stakeholders a clear, consolidated, and actionable understanding of the current architectural landscape.

The following pages provide detailed insights into various aspects of your architecture, including domains, services, messages, ownership, and more.

Key Insights and Report Structure:

- Domains Overview (6 documented):** An overview of business domains, providing context and organization to your services and events.
- Services Overview (8 documented):** Detailed information on all documented microservices, including their versions, summaries, and inter-service communication patterns.
- Messages Overview (37 documented):** A comprehensive listing of all defined events (19), commands (10), and queries (8), their versions, and summaries.
- Channels Overview (3 documented):** Documentation of the messaging topics or channels through which services communicate, providing clarity on data flow and infrastructure dependencies. Documented channels include: kafka, azure-servicebus.
- Teams and Users Overview (27 documented):** Insights into the teams (4) and users (23) documented within the catalog, clarifying ownership, responsibilities, and collaboration points across your architecture.

Architecture Snapshot (as of May 26, 2025)

Metric	Value
Domains	6
Services	8
Messages	37
Channels	3
Teams	4
Users	23

Leveraging this Report:

EventCatalog is committed to empowering your enterprise with the tools for robust architectural governance and strategic evolution. This report serves as a foundational document for identifying areas of improvement, ensuring alignment with business objectives, managing dependencies, and planning future development initiatives. We encourage you to use these insights to foster collaboration, mitigate risks, and enhance the overall efficiency and resilience of your systems.

Governance Score

As of May 26, 2025, the Governance Score for FlowMart provides a composite view of the health and maturity of the documented system components. This score is calculated based on several key metrics, each weighted to reflect its importance.

Overall Governance Score: 75.1 / 100

Score Breakdown by Metric:

Metric	Score (0-100)	Weight (%)	Description
Message Schema Completeness	70.3	22	Percentage of messages (events, commands, queries) with defined schemas.
Service Specification Completeness	25.0	3	Percentage of services with linked API specifications (e.g., OpenAPI, AsyncAPI).
Ownership Coverage	100.0	20	Percentage of resources (domains, services, messages) with assigned owners.
Orphaned Messages	56.8	3	Percentage of messages that have at least one producer.
Service Message Drift	89.1	22	Percentage of services that are sending or receiving the correct version of messages.
Deprecated Messages	62.5	5	Percentage of services that are not using deprecated messages.

Interpreting the Score:

- A higher score (closer to 100) indicates a healthier and more mature architecture according to the defined metrics.
- Review individual metric scores to identify specific areas for improvement.
- Weights can be adjusted to better reflect organizational priorities.

How the Governance Score is Calculated:

The Governance Score is a weighted average of the individual metric scores shown above. Each metric is calculated independently based on data from the EventCatalog, then weighted according to its importance before being combined into the final score.

- Each metric score ranges from 0-100, with higher values indicating better compliance.
- Weights for each metric are assigned based on organizational priorities and architectural importance.
- The final score is the sum of all weighted metric scores, providing a single value from 0-100.

Message Schema Completeness

This section evaluates the **completeness of message schemas** across your architecture. Message schemas are critical for ensuring **reliable communication** between services and preventing **runtime failures** during integration.

Message Schema Score

70.27

26 of 37 messages have schemas

Messages Missing Schemas (11 of 37)

Message schemas are critical for ensuring reliable communication between services. The following messages lack schema definitions, creating potential integration challenges and runtime failures.

Message	Type	Produced By	Consumed By	Owners
User subscription started (v0.0.1)	Event	Subscription Service (v0.0.1)	Payment Service (v0.0.1)	The subscriptions management team
User subscription cancelled (v0.0.1)	Event	Subscription Service (v0.0.1)	Orders Service (v0.0.3)	The subscriptions management team
Payment Processed (v1.0.0)	Event	Payment Service (v0.0.1)	Subscription Service (v0.0.1) Shipping Service (v0.0.1) Notifications (v0.0.2)	David Boyne
Payment Processed (v0.0.1)	Event	Payment Service (v0.0.1)	Subscription Service (v0.0.1) Shipping Service (v0.0.1) Notifications (v0.0.2)	David Boyne
Payment Initiated (v0.0.1)	Event		Payment Service (v0.0.1)	David Boyne
Inventory out of stock (v0.0.4)	Event	Notifications (v0.0.2) Inventory Service (v0.0.2) Inventory Service (v0.0.1)		David Boyne, Martin Smith, Amy Smith, Full stackers, The mobile devs
Inventory out of stock (v0.0.1)	Event	Notifications (v0.0.2) Inventory Service (v0.0.2) Inventory Service (v0.0.1)		David Boyne, Martin Smith, Amy Smith, Full stackers, The mobile devs
Inventory adjusted (v1.0.0)	Event	Inventory Service (v0.0.2) Inventory Service (v0.0.1)	Payment Service (v0.0.1) Orders Service (v0.0.3) Orders Service (v0.0.2) Notifications (v0.0.2)	David Boyne, Martin Smith, Amy Smith, Full stackers, The mobile devs
Inventory adjusted (v0.0.1)	Event	Inventory Service (v0.0.2) Inventory Service (v0.0.1)	Payment Service (v0.0.1) Orders Service (v0.0.3) Orders Service (v0.0.2) Notifications (v0.0.2)	David Boyne
Subscribe user (v0.0.1)	Command		Subscription Service (v0.0.1)	The subscriptions management team

Message	Type	Produced By	Consumed By	Owners
Cancel subscription (v0.0.1)	Command		Subscription Service (v0.0.1)	The subscriptions management team

Recommended Actions:

- **Automated Validation:** Add schema validation to your CI/CD pipeline to ensure compliance
- **Documentation Review:** Schedule regular reviews of schema coverage and quality
- **Owner Accountability:** Ensure clear ownership for each message with missing schemas
- **Add schemas to your message definitions:** Make sure your messages define their schema and the schema is assigned to the message using the `schemaPath` field

Service Specification Completeness

This section evaluates the **completeness of service specifications** across your architecture. Service specifications are essential for **system integration**, **onboarding**, and **compliance efforts**.

Service Specification Completeness Score

25.00

2 of 8 services have specifications

Services Missing Specifications (6 of 8)

The following services require specification documentation to ensure proper integration and governance compliance. Each missing specification represents a potential integration risk and governance gap.

Service	Owners
Subscription Service (0.0.1)	The subscriptions management team
Payment Service (0.0.1)	David Boyne
Shipping Service (0.0.1)	David Boyne
Notifications (0.0.2)	The order management team
Inventory Service (0.0.2)	The order management team
Inventory Service (0.0.1)	David Boyne, Full stackers, The mobile devs

Recommended Actions:

- Add specifications to your services:** Make sure your services define their specifications (OpenAPI or AsyncAPI specs) and the specifications are assigned to the service using the `specifications` field
- Prioritize Documentation:** Focus on documenting specifications for high-visibility or critical services
- Establish Governance Checkpoints:** Implement processes requiring specification documentation before service deployment
- Regular Audits:** Schedule quarterly reviews of service specification coverage
- Team Alignment:** Ensure ownership is clearly defined for services with missing documentation

Ownership Coverage

This section evaluates the **ownership coverage** across your architecture. Ownership coverage is essential for **system integration, onboarding, and governance**.

Ownership Coverage Score

100.00

51 of 51 resources have owners

All resources have owners assigned

Orphaned Messages Score

This section identifies **orphaned messages** within your system. Orphaned messages are messages (events, commands, or queries) that do not have any producers. Identifying and addressing orphaned messages is crucial for maintaining a clean and understandable event-driven architecture, as they can represent outdated integrations, dead code, or incomplete designs.

Orphaned Message Score

56.76

21 of 37 messages have producers

Orphaned Messages (16 of 37)

The following messages do not have any documented producers. These may be candidates for removal or require updates to their producing services.

Message Name	Type	Owners
Payment Initiated (0.0.1)	event	David Boyne
Get subscription status (0.0.2)	query	The subscriptions management team
Get subscription status (0.0.1)	query	David Boyne
Get payment status (0.0.1)	query	David Boyne
Get user notifications (0.0.1)	query	David Boyne
Get notification details (0.0.1)	query	David Boyne
Get inventory status (0.0.1)	query	David Boyne
Subscribe user (0.0.1)	command	The subscriptions management team
Cancel subscription (0.0.1)	command	The subscriptions management team
Update shipment status (0.0.1)	command	David Boyne
Create shipment (0.0.1)	command	David Boyne
Create return label (0.0.1)	command	David Boyne
Cancel shipment (0.0.1)	command	David Boyne
Place Order (0.0.1)	command	David Boyne, Martin Smith, Amy Smith, Full stackers, The mobile devs
Update inventory (0.0.3)	command	David Boyne, Martin Smith, Amy Smith, Full stackers, The mobile devs
Delete Inventory (0.0.3)	command	David Boyne, Martin Smith, Amy Smith, Full stackers, The mobile devs

Recommended Actions:

- **Investigate Orphaned Messages:** Review each orphaned message to determine if it is still relevant or can be deprecated.
- **Document Producers:** If an orphaned message is still in use, update the relevant service documentation to include it in its `sends` array.
- **Automate Detection:** Incorporate checks for orphaned messages into your CI/CD pipeline or regular catalog validation processes.
- **Review Message Lifecycles:** Establish clear processes for message creation, versioning, and deprecation to prevent future orphans.
- **Team Collaboration:** Discuss orphaned messages with the relevant teams to understand their context and ownership.

Service Message Drift

This section identifies services that are sending or receiving **outdated versions of messages**. Keeping message versions up-to-date is important for ensuring compatibility, leveraging the latest features and fixes, and reducing integration risks.

Outdated Message Version Score

89.09%

49 of 55 message interactions use the latest versions

Services Using Outdated Message Versions

The following services are sending or receiving messages that are not the latest version available in the catalog. It is recommended to review these and plan for updates.

Service Name	Message Name	Msg Type	Used Version	Latest Version	Service Owners
Subscription Service	Payment Processed	event	0.0.1	1.0.0	The subscriptions management team
Payment Service	Payment Processed	event	0.0.1	1.0.0	David Boyne
Orders Service	Inventory adjusted	event	0.0.3	1.0.1	The order management team
Orders Service	Inventory adjusted	event	0.0.3	1.0.1	David Boyne
Inventory Service	Inventory adjusted	event	0.0.4	1.0.1	David Boyne, Full stackers, The mobile devs
Inventory Service	Inventory out of stock	event	0.0.3	0.0.4	David Boyne, Full stackers, The mobile devs

Recommended Actions:

- Review Outdated Interactions:** Identify services using older message versions and assess the impact of upgrading.
- Prioritize Updates:** Focus on updating services consuming or producing messages with significant changes or known vulnerabilities in older versions.
- Update Service Configurations:** Modify the service definitions in your EventCatalog to specify the latest message versions once the services themselves are updated.
- Improve Versioning Strategy:** Ensure your teams have a clear strategy for message versioning and a process for communicating updates and deprecations.
- Automate Checks:** Consider adding automated checks to your CI/CD pipelines to flag services using outdated message versions before deployment.

Services Using Deprecated Messages

This section identifies services that are sending or receiving messages that have been **marked as deprecated**. Using deprecated messages can lead to instability, lack of support, and future compatibility issues. It is crucial to migrate away from deprecated components in a timely manner.

Deprecated Message Usage Score

62.50%

Services Using Deprecated Messages

The following services are sending or receiving messages that have been marked as deprecated. It is highly recommended to plan migrations to alternative messages or versions.

Service Name	Deprecated Message	Msg Type	Deprecation Date	Reason	Service Owners
Orders Service	Order amended	event	2025-01-01	N/A	The order management team
Inventory Service	Order amended	event	2025-01-01	N/A	The order management team

Recommended Actions:

- Review Deprecated Usage:** Identify all services interacting with deprecated messages and understand the scope of impact.
- Consult Deprecation Notices:** Review the provided reasons and timelines for deprecation. Contact message owners for clarification or alternatives if needed.
- Plan Migration:** Develop a migration plan to move away from deprecated messages to supported alternatives or newer versions.
- Update Service Definitions:** Once services are updated to no longer use deprecated messages, ensure these changes are reflected in your EventCatalog.
- Monitor Deprecations:** Establish a process to regularly monitor for new deprecations and proactively address them.

EventCatalog Governance Report: Summary & Next Steps

This report provides a comprehensive overview of the current state of your event-driven architecture as documented in EventCatalog for FlowMart. It focuses on key governance metrics to help you identify areas for improvement and ensure the long-term health and scalability of your systems. This summary was generated on May 26, 2025.

Summary of Key Findings

- Overall Governance Adherence: The current overall governance score is 75.1/100. Specific areas are detailed below.
- Detailed findings on schema completeness, service specifications, ownership, orphaned messages, message version drift, and deprecated message usage will be analyzed based on current data.

Top 3 Governance Priorities

- Enforce Schema and Specification Completeness:** Mandate and automate the validation of complete, formal schemas (e.g., JSON Schema, Avro) for all new and updated messages. Ensure all services have accurate and current AsyncAPI (for event-driven services) or OpenAPI (for request/response services) specifications. **Impact:** Reduces integration errors, improves data quality, and enhances service discoverability.
- Establish and Maintain Clear Ownership:** Assign explicit owners (teams or individuals) to every domain, service, event, and command within EventCatalog. Implement a review process to keep ownership information current. **Impact:** Ensures accountability, streamlines communication, and facilitates quicker decision-making and issue resolution.
- Manage Message Lifecycle Actively:** Regularly review and address orphaned messages. Develop and enforce a clear strategy for versioning and deprecating messages, including a plan for migrating consumers off deprecated versions. **Impact:** Reduces system clutter, minimizes risks associated with outdated contracts, and ensures a healthier, more maintainable event ecosystem.

What's Next?

- CI/CD Integration for Governance Checks: Embed automated governance checks into your CI/CD pipelines.
- Runtime Metrics Correlation: Explore integrating design-time governance insights with runtime operational metrics.
- Governance Score Trend Analysis: Track key governance metrics over time to measure improvement.
- Enhanced Developer Experience: Leverage EventCatalog data to generate boilerplate and SDKs.
- Automated Notifications & Nudges: Implement systems to notify teams about relevant governance issues.

Disclaimer

This EventCatalog Governance Report is based on the design-time metadata available within your EventCatalog instance as of the generation date. It focuses on the structural, definitional, and relational aspects of your event-driven architecture, including schema completeness, service specification coverage, ownership, and message lifecycle indicators like orphaned messages, version drift, and deprecated message usage. **This report does not incorporate real-time operational metrics, runtime performance data, security vulnerability assessments, or compliance**

with specific regulatory standards unless explicitly stated as part of a custom integration. The findings and recommendations are intended to guide strategic improvements in your EDA governance practices.