

# Data Coverage: A Comprehensive Approach

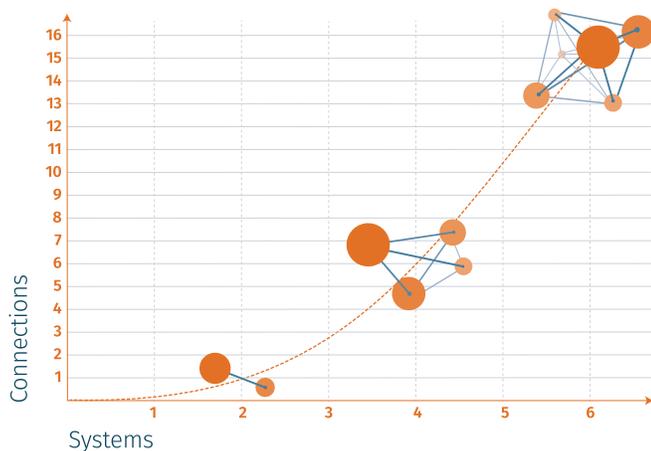
## Case Study Series | Part 2

*This is the second in a series of discussion papers outlining the core design challenges 3Forge has addressed. In this discussion, we focus on integration with fragmented data.*

Data fragmentation is the result of an evolutionary process: Companies must remain competitive and compliant so they consistently reinvest in new technologies. As they progress through this perpetual tech cycle, increased system diversity and layers of technology debt result in overwhelming data fragmentation. There are many solutions that focus on a particular aspect of this problem but 3Forge has designed and developed a hybrid approach that provides true ubiquitous access.

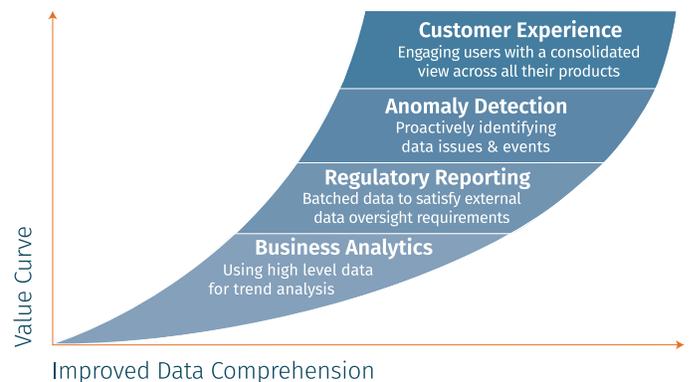
## Geometric Complexity Growth

This adoption of new technologies introduces an overwhelming challenge: A geometric increase of data flow and a far more complex data ecosystem.



## Value Curve

This a critical topic and solving the consolidated access to distributed data unlocks huge organizational benefits.



The challenges of data consolidation and reconciliation go far beyond data storage:

- **Schema Changes**  
Types, labels & structures continually change.
- **Version Contagion**  
Systems are updated in phases, meaning inconsistent versions and protocols.
- **Volatile Ecosystem**  
Individual system interruptions and updates.
- **Real-time Streaming Data Bursts**  
Bursts of data can have fatal down-stream impacts.
- **Hard to Access Data**  
Local files and operating system resources are often out of reach.

# The Evolution of Data Access

## The Data Lake

*Simply duplicate all the data into one location.*

Fundamentally, this suffers from a monolithic architecture and high maintenance ETL. It only solves a small piece of the problem, it's inefficiently rigid and doesn't address the challenges outlined above.

## A Federated Database

*A single "virtual database" that doesn't store data directly but instead logically routes queries to other data sources on demand.*

This is a more distributed approach and can even leverage existing data lakes. It provides a unified access point but the direct interaction between federated databases and their underlying data sources still suffers from challenges outlined above.

## Distributed Data Access Layer

*A layer of semi-distributed microservices dedicated to blending the diverse nature of systems into a consolidated format for access.*

It can route real-time data and dispatch on-demand queries. This layer provides comprehensive fault-tolerant data access using a dynamic design which is able to handle unpredictable future requirements without requiring system-wide updates.

# The 3Forge Distributed Relay Architecture

## Fast, Reliable Data Consumer

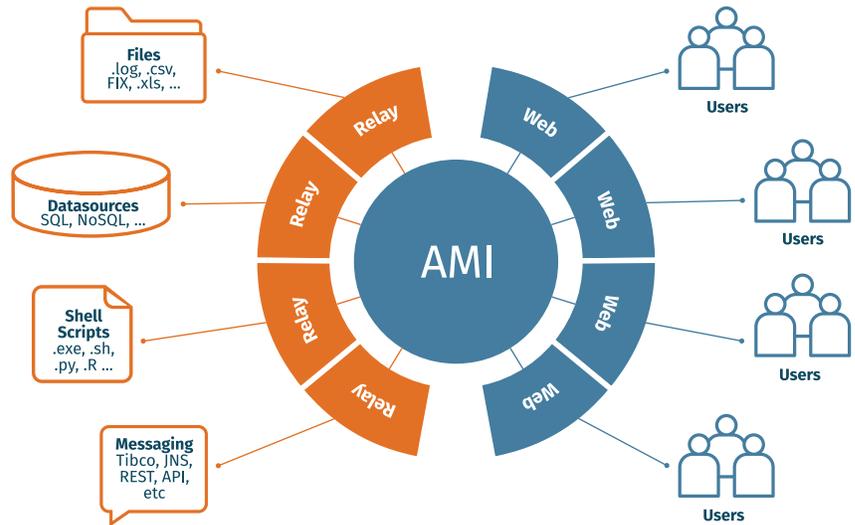
Real-time data communication is error prone. A chief concern is slow consumption, forcing the mission critical producer to back up.. Placing the high speed AMI Relay near the producer of real time data shortens the distance and provides a fault-tolerant buffer.

## Portability

Interfacing with multiple incompatible versions of a database, operating system, or non-interchangeable protocol are solved using individually configured AMI Relays, mapping the turbulent environment into a consistent, accessible protocol.

## Localized Audit & Entitlements

Understanding the logical rules for who should have access to data is often best understood by the localized team managing that data. An additional localized entitlement layer allows for sophisticated, granular data permissions to be done in a distributed fashion.



## Immunization

Server processes go down, network connections get lost, third party adapters crash and hardware fails. But the otherwise systemic impact stops at the AMI Relay, keeping outages localized and limited, providing a best case scenario.

## Access to Everything

Files, executables and O.S. functionality are often not readily reachable. By installing an AMI Relay locally, these resources now become remotely accessible and monitorable.

# The AMI Relay Architecture

## Realtime Streaming Store & Forward Guaranteed Messaging

Rapidly consumes data off the wire and journals to disk. Late joining or slow consuming centers will be fed data from the journal. This journal is valuable for replay and audit purposes.

## Realtime Feed Handlers

Converts real time data from source-specific formats to AMI's highly efficient protocol. These messages are validated, compressed and dispatched to the center(s).



## Partitioning

Configured to route messages across centers according to their content. This enables multiple centers to process data in parallel.

## Entitlements/Audit Layer

Controls what data is permitted vs restricted and produces a detailed "who did what & when" journal across the enterprise.

## Data Virtualization

Each Relay has its own micro-federated database. When it receives a query, the request is dispatched to the appropriate datasource adapter for execution.

## Final Thoughts

The 3Forge AMI Relay is a distributed solution which provides comprehensive integration, regardless of form or function. While most solutions focus on a specific use case, only this hybrid approach makes ubiquitous data access a reality.

### Discovery

Will automatically detect and make a relay's resources available.

### Built in Data Federation

AMI automatically routes requests to appropriate data sources

### Data Lake Integration

Connect to existing data lakes for immediate access to centralized data.

### ETL

Extract/Transform/Load data across relays.

### Data Warehouse

Store commonly used data inside AMI for rapid access.

### On Demand

Access to data as users, timers or events need it.

### Streaming

Relays capture realtime streaming data for instant use access and analytics.

### Macro Execution

Send events on a message bus, call stored procedures or execute scripts.

### Safe & Secure

Entitlements and audit at various checkpoints including at the relay level.