

# DMaaP – Data Movement as a Platform proposal

## Project Name:

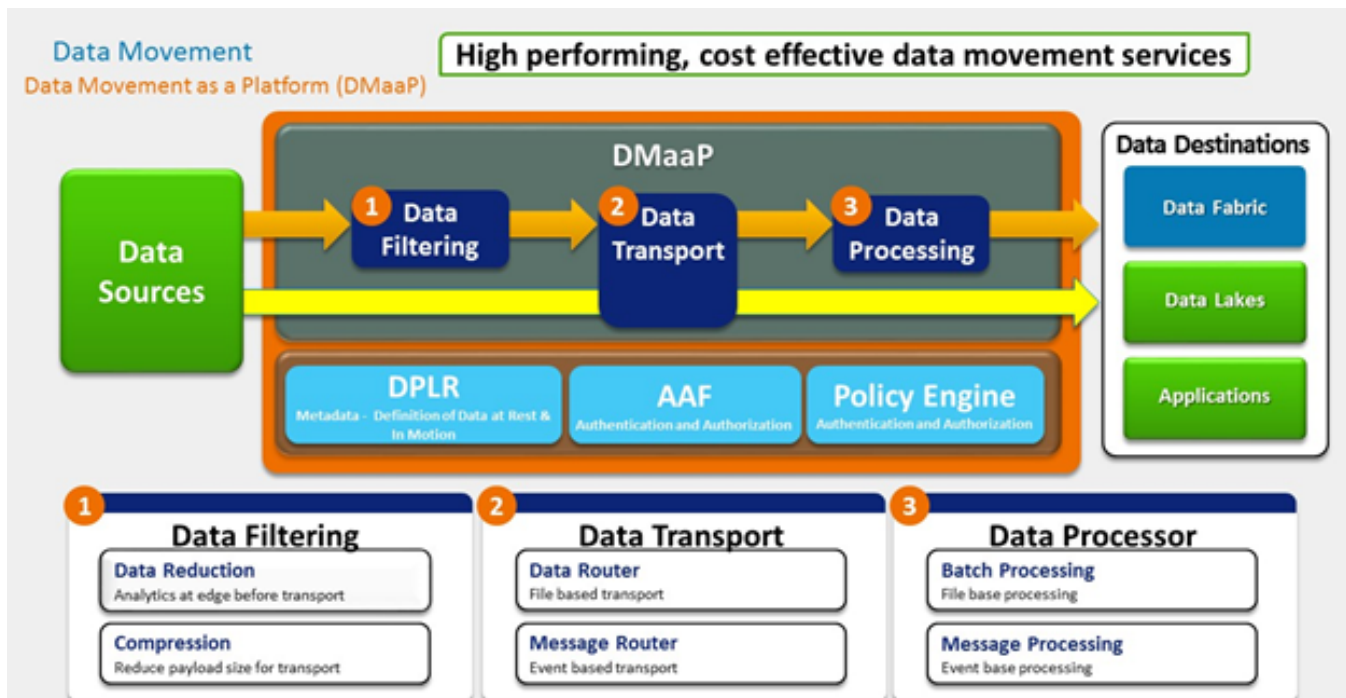
- Proposed name for the project: DMaaP – Data movement as a platform
- Proposed name for the repository: DMaaP

## Project description:

DMaaP is a premier platform for high performing and cost effective data movement services that **transports** and **processes** data from any source to any target with the format, quality, security, and concurrency required to serve the business and customer needs.

DMaaP consists of three major functional areas:

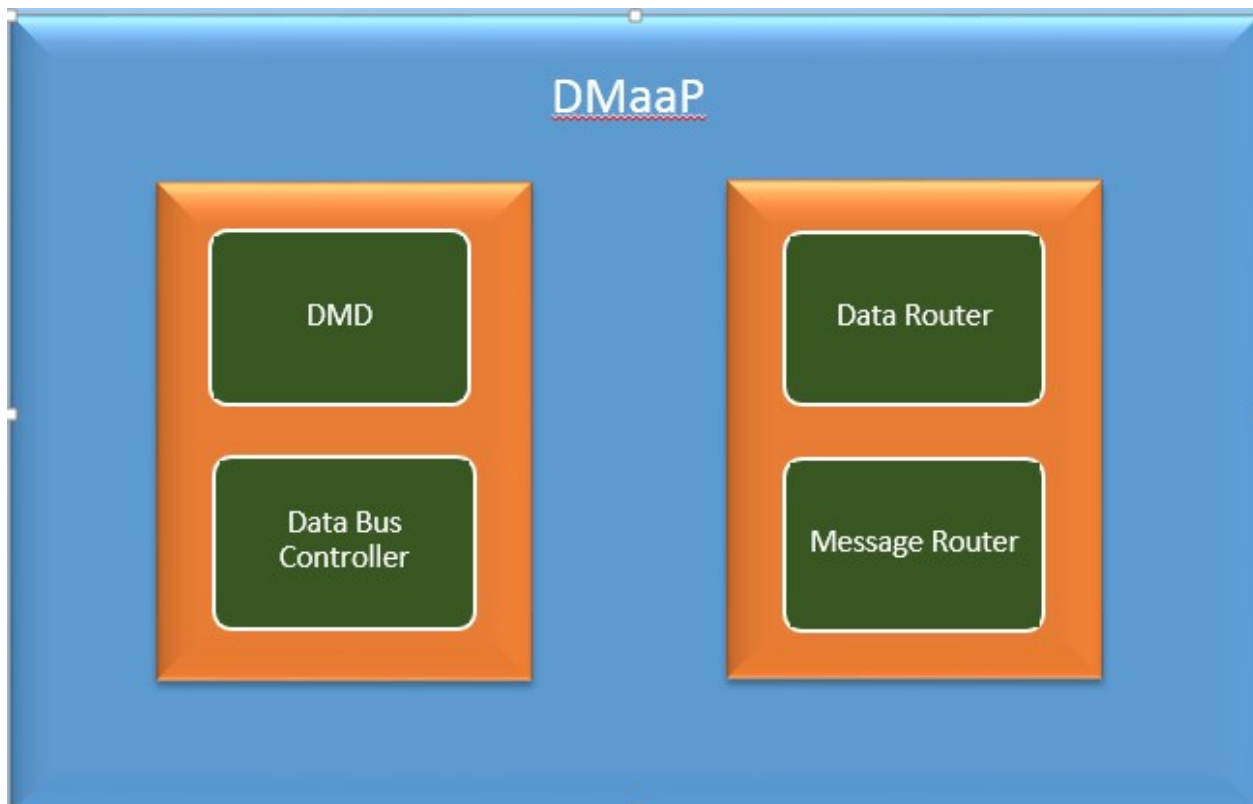
- 1. Data Filtering** - the data preprocessing at the edge via data analytics and compression to reduce the data size needed to be processed.
- 2. Data Transport** - the transport of data intra & inter data centers. The transport will support both file based and message based data movement. The Data Transport process needs to provide the ability to move data from any system to any system with minimal latency, guaranteed delivery and highly available solution that supports a self-subscription model that lowers initial cost and improves time to market.
- 3. Data Processing** - the low latency and high throughput data transformation, aggregation, and analysis. The processing will be elastically scalable and fault-tolerant across data centers. The Data processing needs to provide the ability to process both batch and near real-time data.



## Scope:

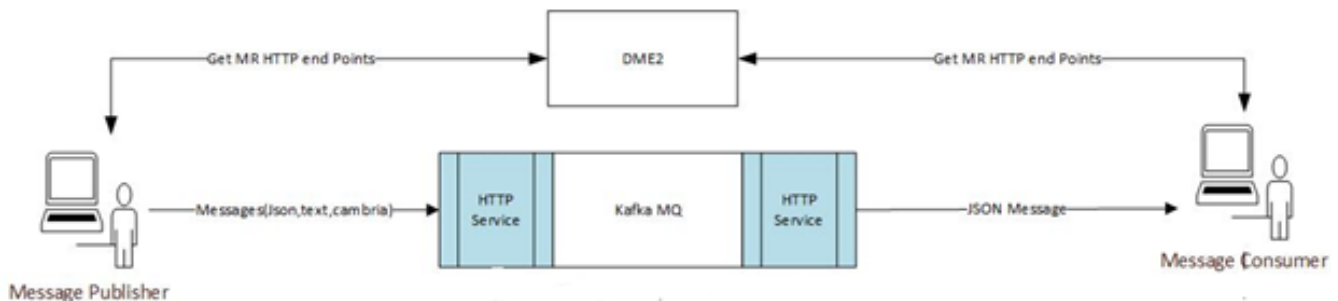
DMaaP has four components:

- 1. Message Router (MR)** - Message Router is a reliable, high-volume pub/sub messaging service with a RESTful HTTP API. It is intended to be deployed by Platform Service providers so that it is available to Platform clients as a web service. The service is initially built over Apache Kafka.
- 2. Data Router (DR)** - The Data Routing System project is intended to provide a common framework by which data producers can make data available to data consumers and a way for potential consumers to find feeds with the data they require. The interface to DR is exposed as a RESTful web service known as the DR Publishing and Delivery API
- 3. Data Movement Director (DMD)** - A client to DMaaP platform to publish & subscribe data.
- 4. Data Bus Controller** - Provisioning API of the Data Movement Platform.



#### Message Router:

In DMaaP Message Router, Restful web service is exposed to client to perform any needed action with Kafka. After getting the request it calls the Message router service layer which is created using AJSC ( AT&T Java Service Container) . AJSC finally calls Kafka services and response is sent back.



#### Message Router will include the following functionality:

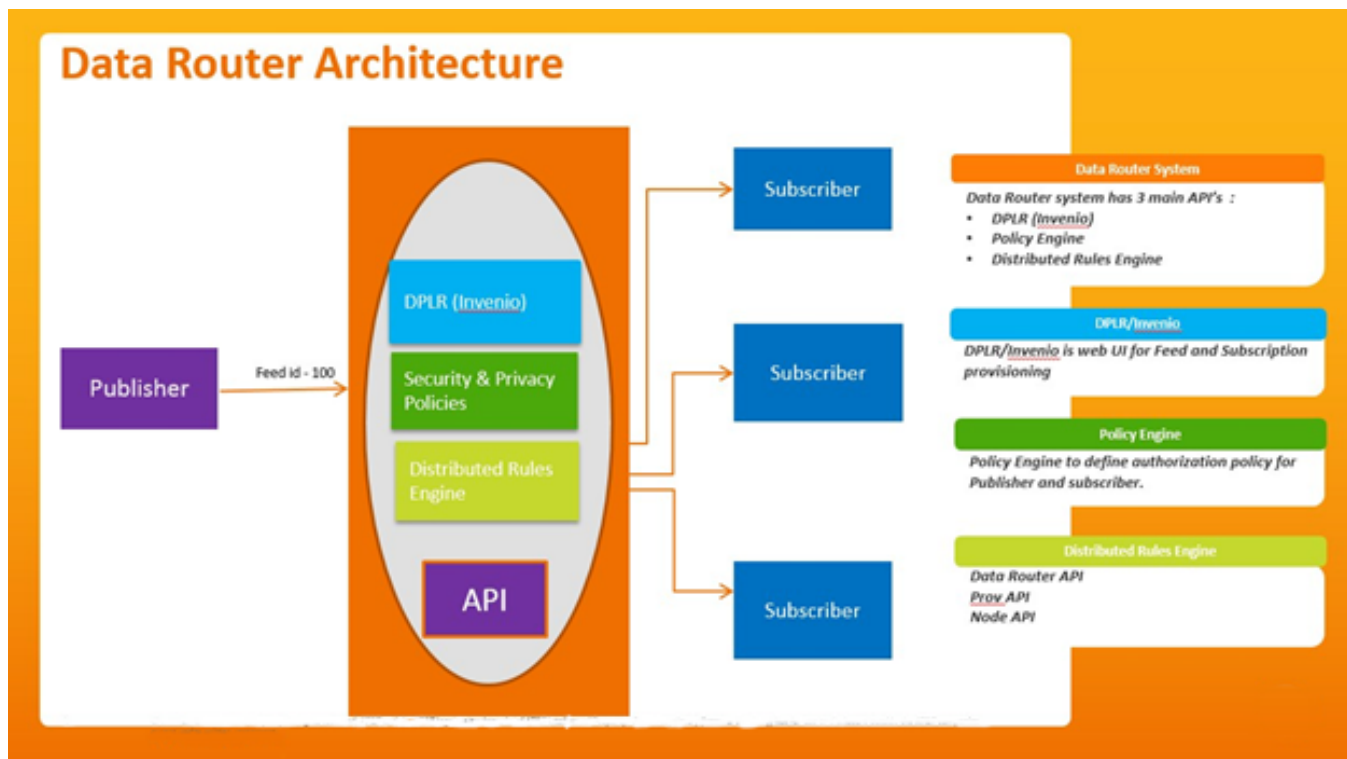
- Pub/sub messaging metaphor to broaden data processing opportunities. (refers to the label of the rendezvous point as a "topic")
- A single solution for most event distribution needs to support a range of environments.
- Packaged to support deployment configurations ranging from a single container to a cluster of hosts.
- Horizontal scalability: Add servers to the cluster to add capacity.
- Durability: Hardware failure in the cluster should not impact service, and messages should never be lost.
- Durability: Consumers should not lose messages if they experience downtime.
- High throughput: Consumers must be able to distribute topic load across multiple hosts in a cluster.
- Easy integration via RESTful HTTP API:
  - Implements a RESTful HTTP API for provisioning
  - Implements a RESTful HTTP API for message transactions (i.e. pub, sub)
  - Implements a RESTful HTTP API for transaction metrics
- Optionally registers with supporting network locations services which will allow client connections to nearest end point. (when there are multiple deployments in a network)
- Client authentication and authorization models:
  - Supports multiple authentication and authorization models;
  - Default model is "insecure" (i.e. not authenticated)
  - Model is associated per topic or per some group of topics.

- Optionally supports a model that relies on external service for client authentication and authorization.
- Optionally supports replication of messages for selected topics to another instance of DMaaP MR.
- Optionally supports different underlying message bus technologies, but makes this transparent to clients.
- Standardized topic names
- Topic registry and discovery
- Recover partially delivered messages from the point of failure

**Data Router** will include following functionality.

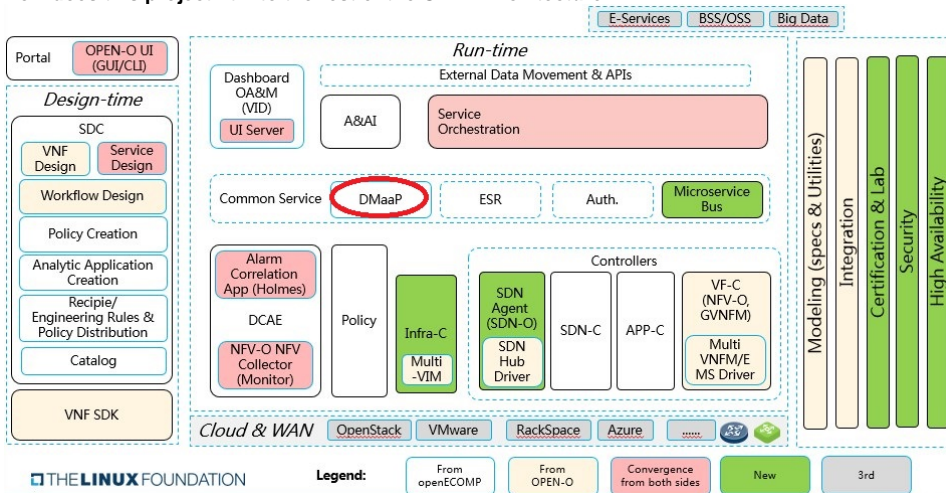
- It is based on Pub/Sub architecture, not Point to Point. Adding new subscribers (targets) for existing published feed requires no work on the publisher (Source side)
- Agnostic to source and sink of data residing in an RDBMS, NoSQL DB, other DBMS, Flat Files, etc.
- Does not tightly couple publisher and subscriber endpoints.
- Ability to track completed file transmissions (via log records)
- Using HTTP as File transfer Protocol allows customers to utilize the language and frameworks that they are most comfortable with.
- Does not require a contract (connectivity configuration) between two servers (not server specific Point to Point). It's simply a contract between two applications.
- Automatically retries delivery to subscriber for 24 hours. If it still fails, it will be marked as permanently failed.
- User authentication for each publish request.
- Authorization for each end point (IP of publisher) determines if allowed to publish or not.
- Supports low latency for transmission of large files.
- Feed can be defined with a guaranteed delivery option. This includes dealing with subscribers that are down for an extended amount of time.

**Data Router:**



**Architecture Alignment:**

- How does this project fit into the rest of the ONAP Architecture?



- What other ONAP projects does this project depend on?
  - AAF
- Are there dependencies with other open source projects?
  - JSC ( Java Services Container )
  - DME ( Direct Messaging Engine )
  - AAF ( Application Authorization Framework )
  - Apache Kafka
  - Zookeeper.
  - Apache Tomcat

## Release 1:

Release 1 provides the following features:

- Pub/sub messaging metaphor to broaden data processing opportunities.
- A single solution for most event distribution needs to simplify our solutions.
- Horizontal scalability: Add servers to the cluster to add capacity. (Our initial installations are expected to handle at least 100,000 1KB msgs/sec per data center.)
- Durability: Hardware failure in the cluster should not impact service, and messages should never be lost.
- High throughput: Consumers must be able to distribute topic load across multiple systems.
- Easy integration via RESTful HTTP API

## Resources:

- Primary Contact Person - **Bhanu Ramesh (ATT), Ram Koya (ATT), John Murray(ATT), DOMINIC LUNANUOVA (ATT)**
- Names, gerrit IDs, and company affiliations of the committers :
- Names and affiliations of any other contributors
- Project Roles (include RACI chart, if applicable)

Name	Gerrit ID	Company	Email	Time Zone
Bhanu Ramesh		AT&T	bg6954@att.com	USA, EST
Ram Koya		AT&T	rk541m@att.com	Dallas, USA CST/CDT
John Murray		AT&T	jfm@research.att.com	Bedminster, USA EST /EDT
Dominic Lunanuova		AT&T	dgl@research.att.com	MiddleTown, USA EST/EDT

## Other Information:

- link to seed code (if applicable): <https://github.com/att/dmaap-framework>
- Vendor Neutral
  - The current seed code has been already scanned ( Using Fossology and Blackduck) and cleaned up to remove all proprietary trademarks, logos.
  - Subsequent modification to the existing seed code should continue to follow the same scanning and clean up principles

## Key Project Facts

### Project Name:

- JIRA project name: Data Movement as a Platform
- JIRA project prefix: DMAAP-

**Repo name:** DMaaP

**Lifecycle State:** incubation

**Primary Contact:** Bhanu Ramesh (PTL Delegate) / Ram Koya

**Project Lead:** Bhanu Ramesh (PTL Delegate) / Ram Koya

**mailing list tag** [Should match Jira Project Prefix]

### Committers:

Ram Koya [rk541m@att.com](mailto:rk541m@att.com)

Varuneshwar Gudisena [vg411h@att.com](mailto:vg411h@att.com)

Habib Madani [habib.madani@huawei.com](mailto:habib.madani@huawei.com)

Xinhui Li [lxinhui@vmware.com](mailto:lxinhui@vmware.com)

### Contributors:

Ramdas Sawant [rs873m@att.com](mailto:rs873m@att.com)

Bhanu Ramesh [bg6954@att.com](mailto:bg6954@att.com)

Ramkumar Sembaiyan [rs857c@att.com](mailto:rs857c@att.com)

Ramdas Sawant [rs873m@att.com](mailto:rs873m@att.com)

Vikram Singh [vs215k@att.com](mailto:vs215k@att.com)

Sai Gandham [sg481n@att.com](mailto:sg481n@att.com)

Dominic Lunanuova [dgl@research.att.com](mailto:dgl@research.att.com)

Catherine Lefèvre [cl664y@intl.att.com](mailto:cl664y@intl.att.com)

### Link to approval of additional submitters: