

# Schema element workgroup update

David Radley (Egeria maintainer @IBM) and Jürgen Hemelt (Atruvia)

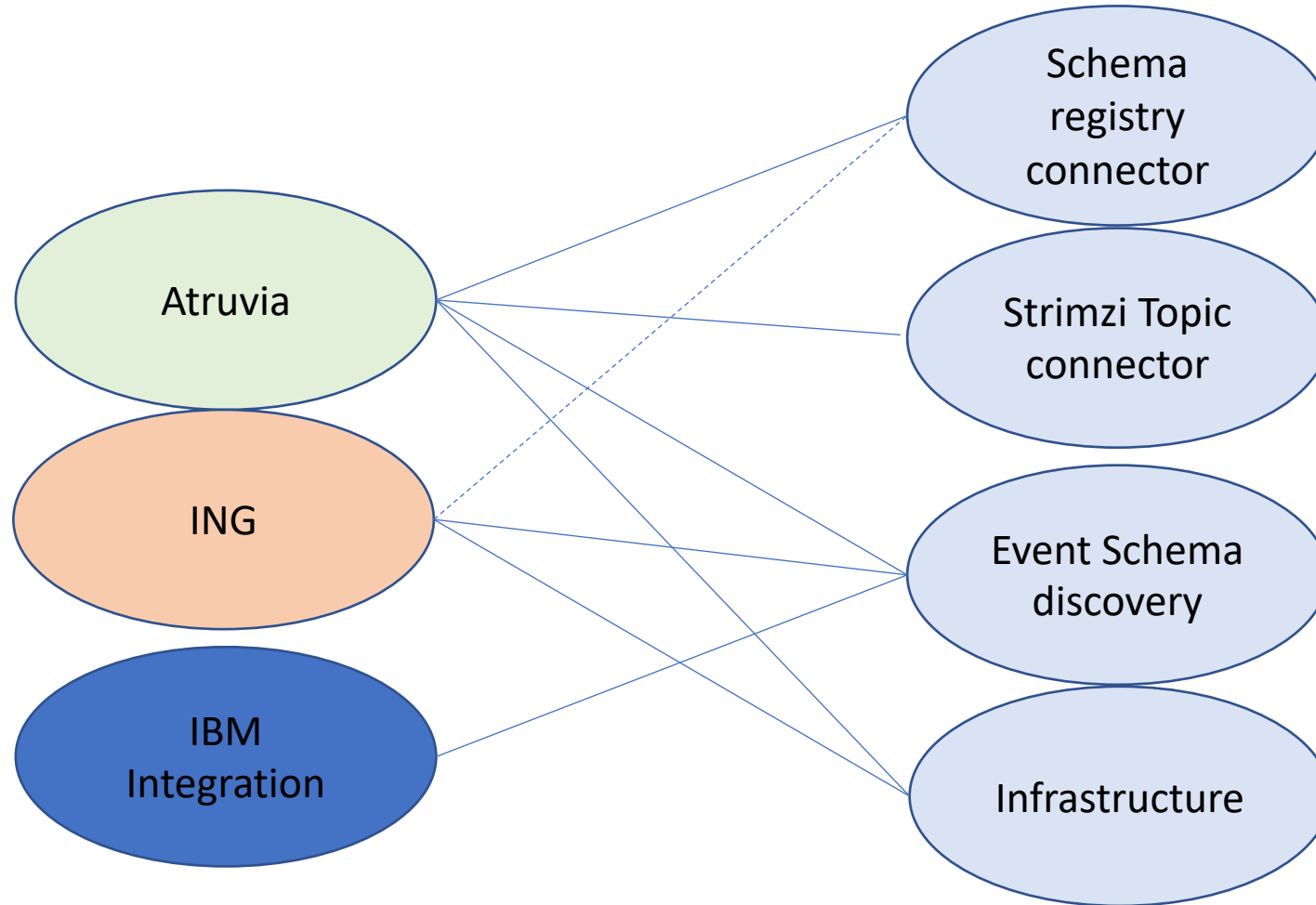
# Workgroup participants

- IBM
  - David (Egeria)
  - Nigel (Egeria)
  - Nic Townsend (IBM Integration)
  - Ian Perry (IBM sales)
- Atruvia
  - Jürgen Hemelt
  - Darius Jockel
  - Marcel Engbers
  - Tunç Taylan Turunç
- ING
  - Ljupcho Palashevski

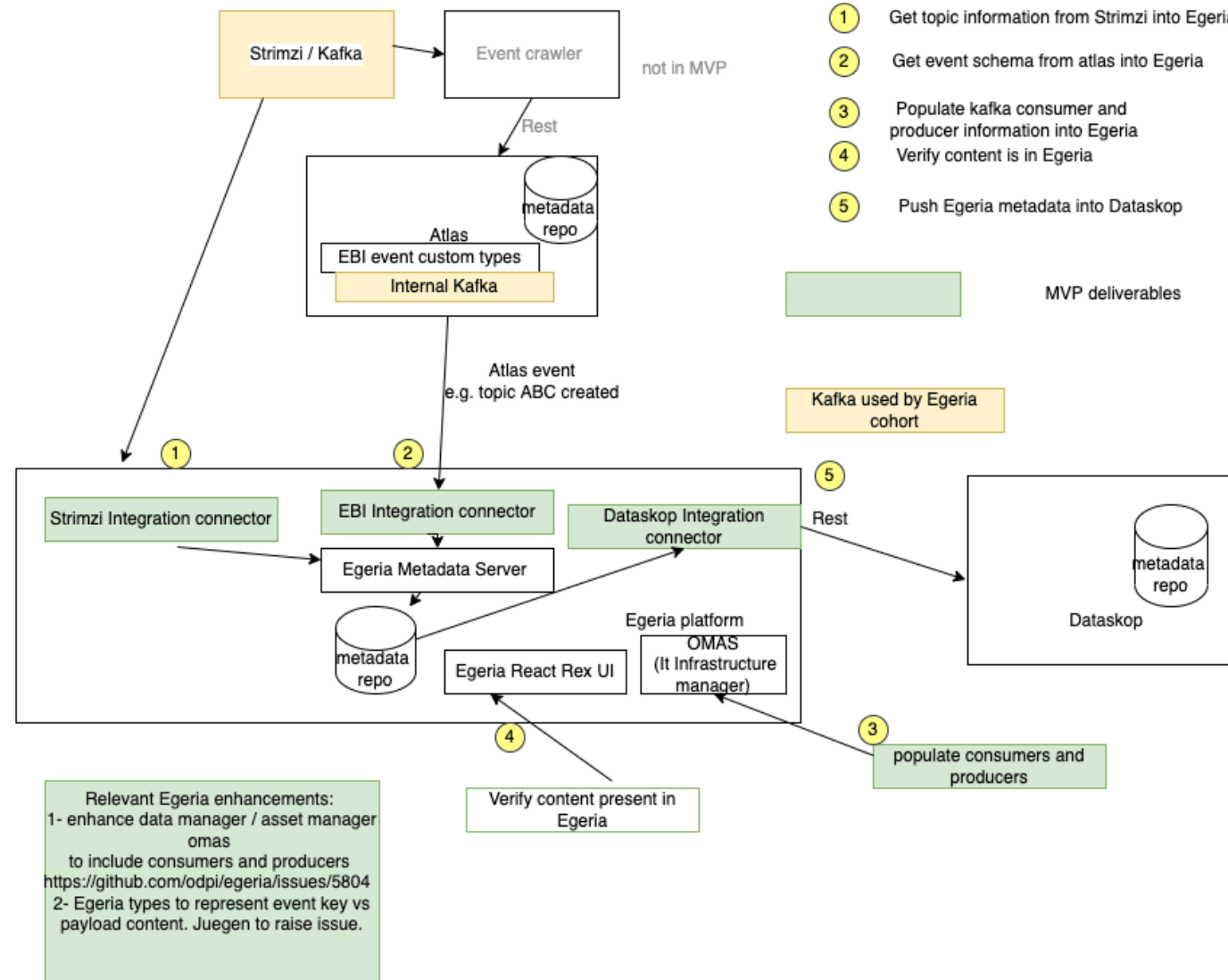
# Meeting schedule

- 9<sup>th</sup> March
- 17<sup>th</sup> March
- 5<sup>th</sup> May Next meeting

# Priority common requirements



# Background Atruvia MVP topology



# New Atruvia requirements

- Drop Atlas, use an Egeria connector
- Support more event formats for discovery plain json and maybe protobuf
- Get schema from:
  - Introspection of event payloads
  - Confluent schema registry
  - Other schema registries

# Priorities

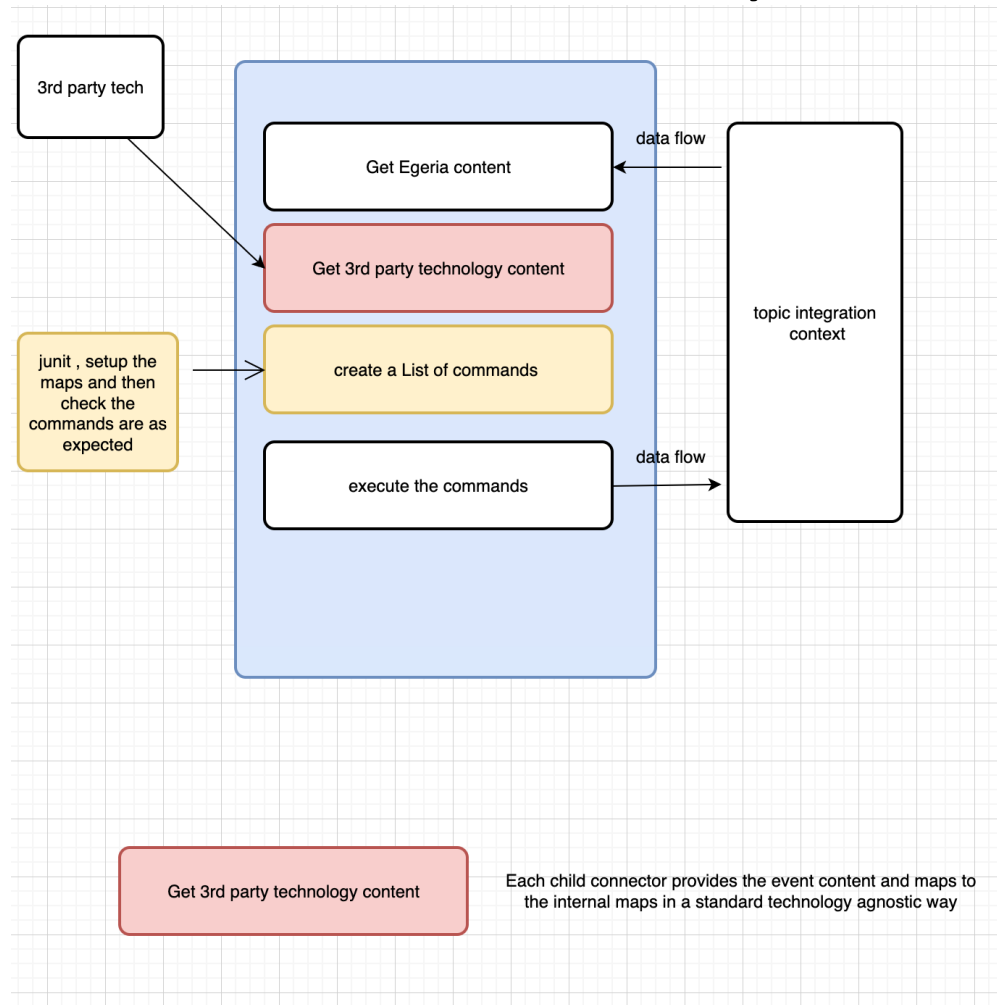
- Juergen presented his vision for Atruvia and mentioned:
  - A Data Catalog connector
  - An OPA Connector which will determine visibility and access and the like
  - The following order is roughly the priority order of integration connectors they are interested in developing
    - Strimzi topic
    - Event schema (Confluent Schema Registry and others)
    - Discovery processing
    - Strimzi infrastructure
    - OPA
    - Data Catalog
  - We mentioned the producers and consumers, there was a question around where exactly these APIs should be called and so these APIs can be supplied with the appropriate information.
- Ljupcho and Nic talked of being interested in the discovery part of this.

# Workgroup initial achievements

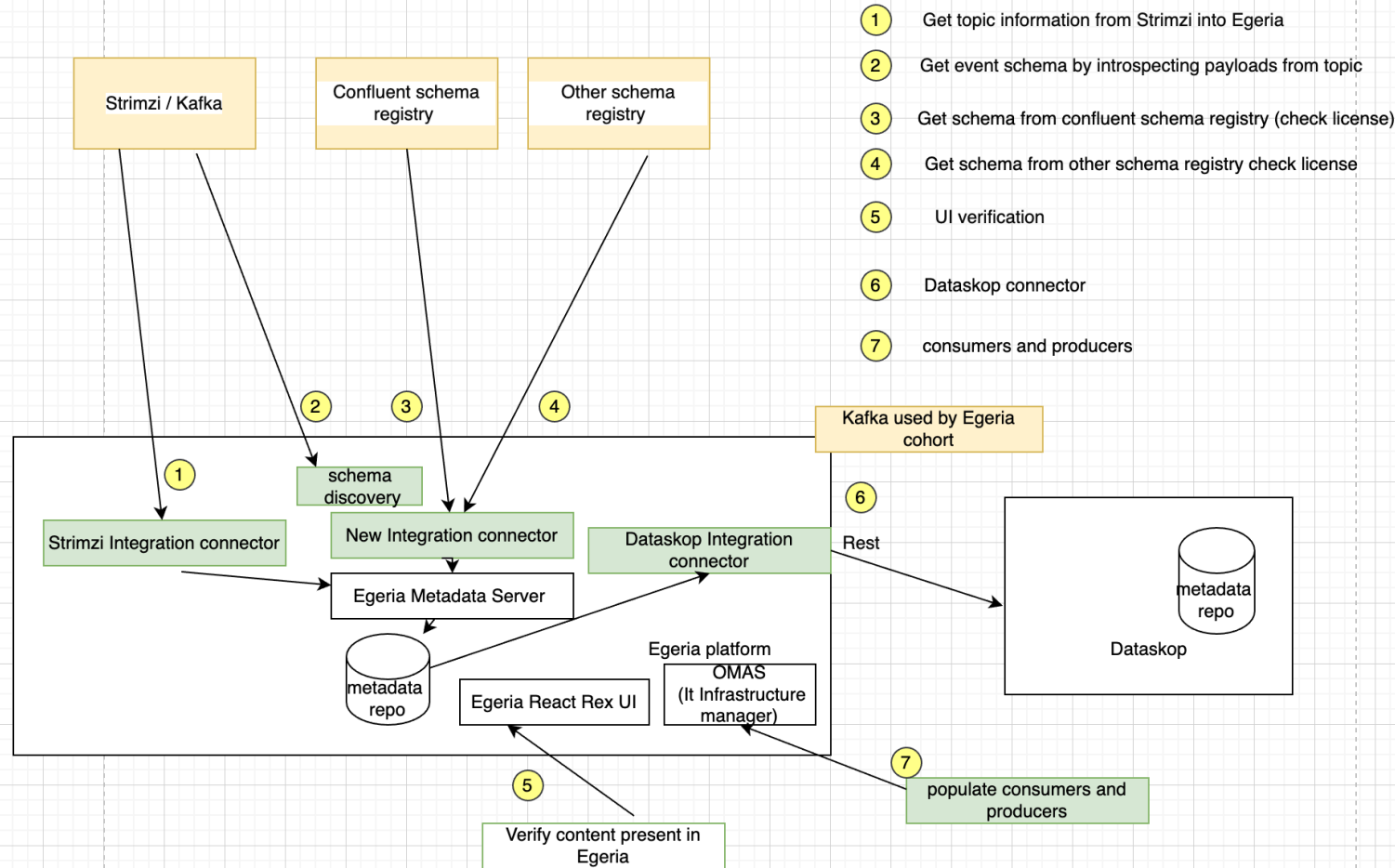
- We created a new Egeria repository for the Strimzi connector
  - Populated and builds
  - <https://github.com/odpi/egeria-connector-integration-topic-strimzi>
- We create a new Egeria repository for the event schema connector
  - Not yet populated, will hold schema registry connector
  - Ongoing Atruvia development in their current internal dev sprint.



# More details on the event schema integration connector – ‘in memory’



# A Proposed 'all in memory' topology



# More sophisticated topologies

- Minimize memory footprint, using events do minimal processing:
  - For small schemas – polling all in memory might be quicker as there is no re-querying the same elements
  - Is there really a schema that would be too big for in-memory?
    - Scalability testing required to further understand how different connector patterns effect the overall time to process changes?
  - If changes to schemas are few and minimal – grabbing the whole schema seems overkill / inefficient.

# Discovery

- This discovery should not occur in the integration connector; it should use:
  - engine host
  - ODF (discovery framework) .
- Talking of looking at plain json event payloads , and extracting String & list & object names relationships and properties into Egeria.
- Possibly consume protobuf

# Infrastructure OMAS + Integration connector

- Using this to pull out the Kafka cluster and Kafka Connect cluster configuration from Kubernetes
- Use Strimzi custom resource definitions
- Link to Kafka Topics and Kafka Connectors

Current Status Atruvia

# First version of the Event Schema Connector

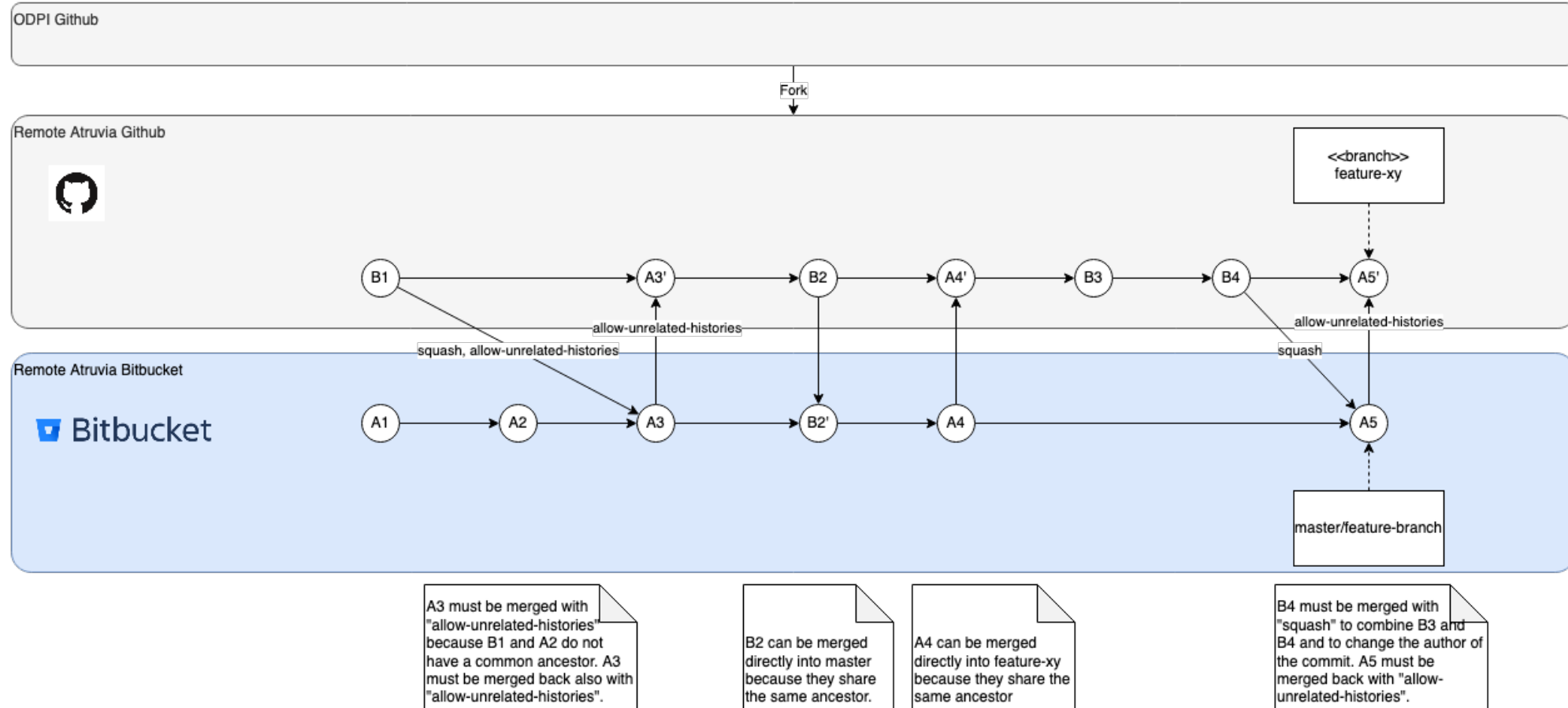
- Take all top level event schemas from Egeria into a local cache
- Read all subjects from Schema Registry and extract the topic name
- Get the latest version of the subject and compare it against the local cache
- If the latest version is newer than the cache, replace the schema including all sub-schemas

# Integrate Github and Atruvia Bitbucket

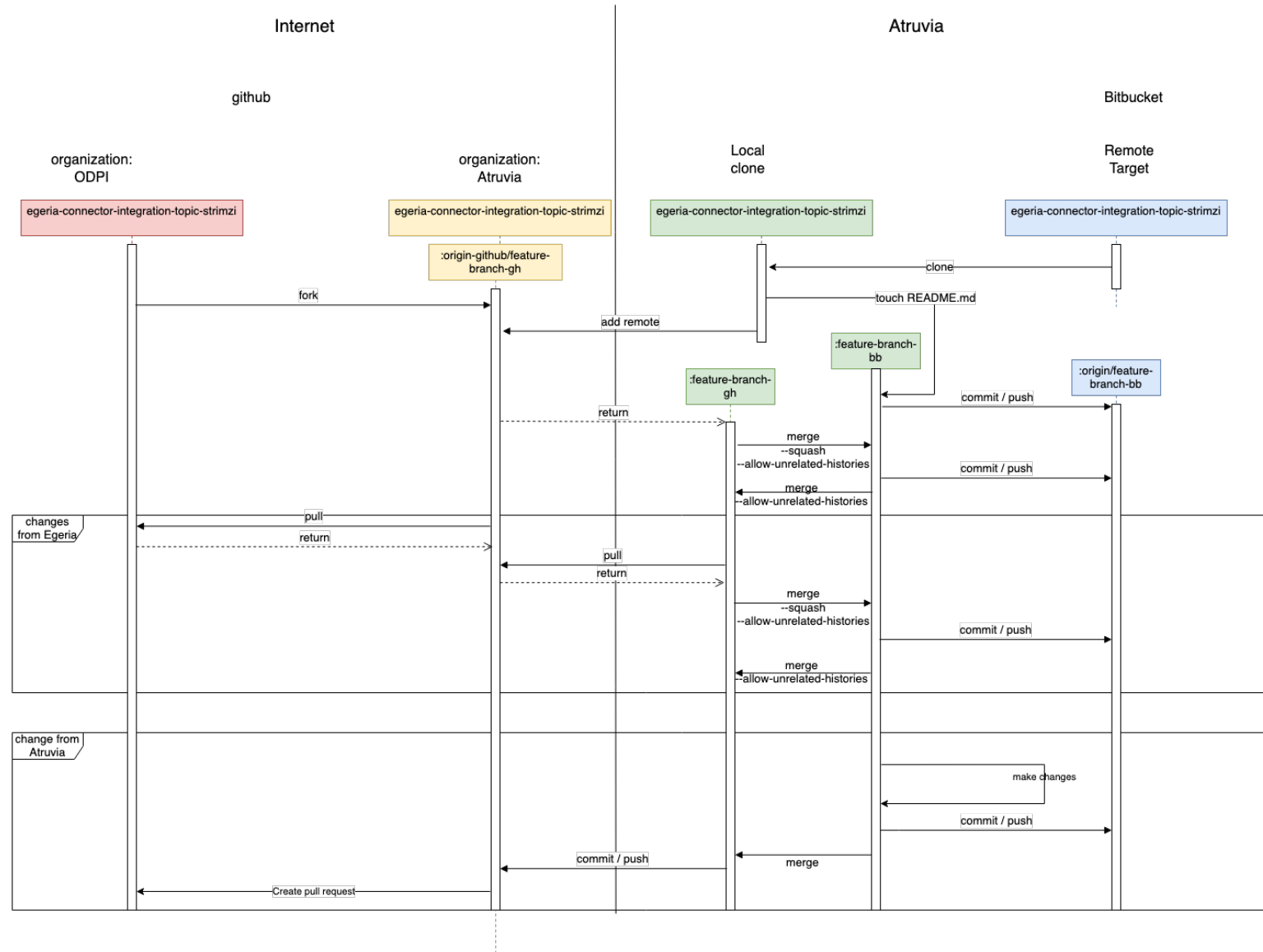
- Atruvia uses Bitbucket as code repository
- All code has to be maintained in Bitbucket
- Only Atruvia employees are allowed to commit to Bitbucket
- Hence, code from Github cannot be pulled directly into Bitbucket (unknown committers)



# How we can handle this



# A more detailed view ...



# More challenges

- Pushing changes from Bitbucket to Github makes personal data of all committers public (e-mail addresses) → must be accepted from each committing employee (GDPR regulations)
- No complete changelog in Bitbucket → regulatory requirements