Timeseries Event Channel

Get insight into your energy usage within hours via AgileDX

2024

CGI

Summary

Datahub

- Business Context
- System Context

Background

- Timeline
- Legislation change
- NextGen meter
- Datahub characteristics
- Stakeholder survey
- Focus Area Technology Study

Technology Study

- Approach
- Patterns
- Requirements
- Technology Analysis
- Design

Proof of Concept

Advise on proof of concepts to be done in phase 2 to determine the feasibility of the chosen architecture and its components.

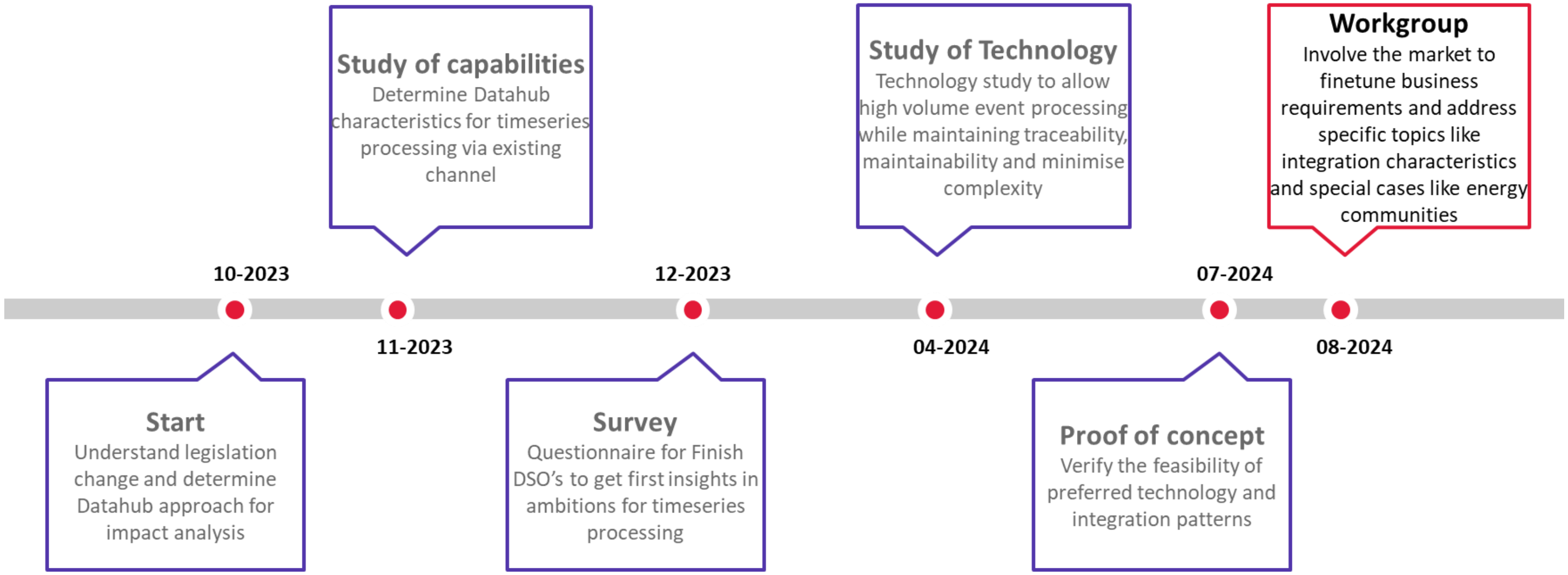
Summary

Summary

The decree 767/2021 “Valtioneuvoston asetus sähköntoimitusten selvityksestä ja mittauksesta” (Government Decree on the Settlement and Measurement of Electricity Supplies) introduces a directive to make measurement data available within 6 hours per 1/1/2026.

A preliminary analysis of what this could mean for involved parties and the Datahub has been performed and the verification on the feasibility of a high performant event channel has started.

Next step is to closely involve the market to identify the business needs, align on interpretation, align on technicalities including specifications and working models and produce acceptable solutions for affected functionalities.



10-2023

Start
Understand legislation change and determine Datahub approach for impact analysis

11-2023

Study of capabilities
Determine Datahub characteristics for timeseries processing via existing channel

12-2023

Survey
Questionnaire for Finish DSO’s to get first insights in ambitions for timeseries processing

04-2024

Study of Technology
Technology study to allow high volume event processing while maintaining traceability, maintainability and minimise complexity

07-2024

Proof of concept
Verify the feasibility of preferred technology and integration patterns

08-2024

Workgroup
Involve the market to finetune business requirements and address specific topics like integration characteristics and special cases like energy communities

Introduction Datahub

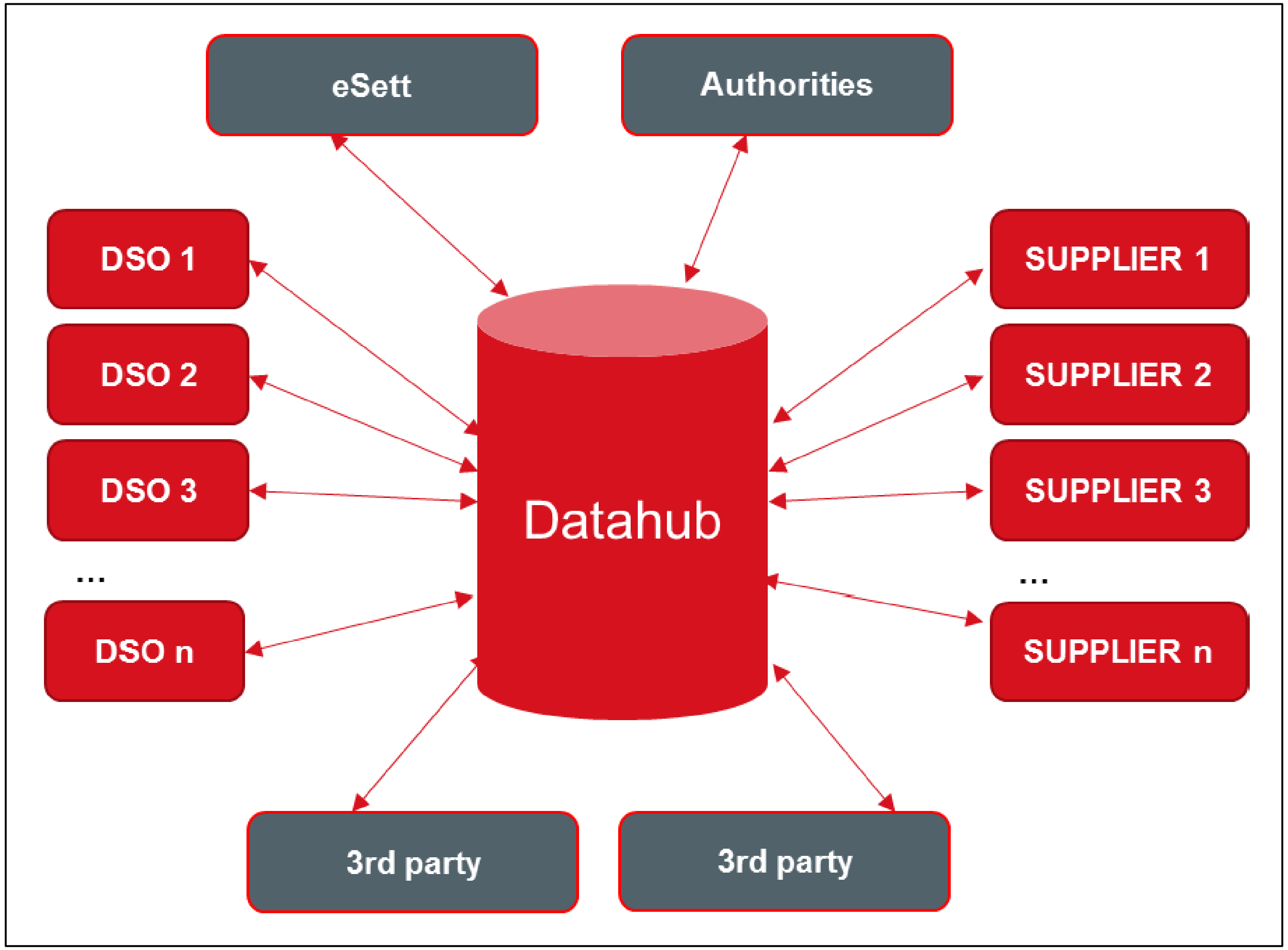
Business Context

Datahub Introduction

Amended in 2013, the Electricity Market Act gave Fingrid Oyj (Fingrid) responsibility for developing the information exchange required for electricity trade and imbalance settlement; the Fingrid Datahub.

Since its implementation, electricity retail market information exchange in Finland is based on a centralised model, in which information exchange between electricity market parties takes place via the Datahub and information is saved in a centralised data storage.

The implementation of this model will trigger a transition from the existing batch processing-based asynchronous information exchange model to real-time synchronous information exchange.



System Context

Introduction Datahub

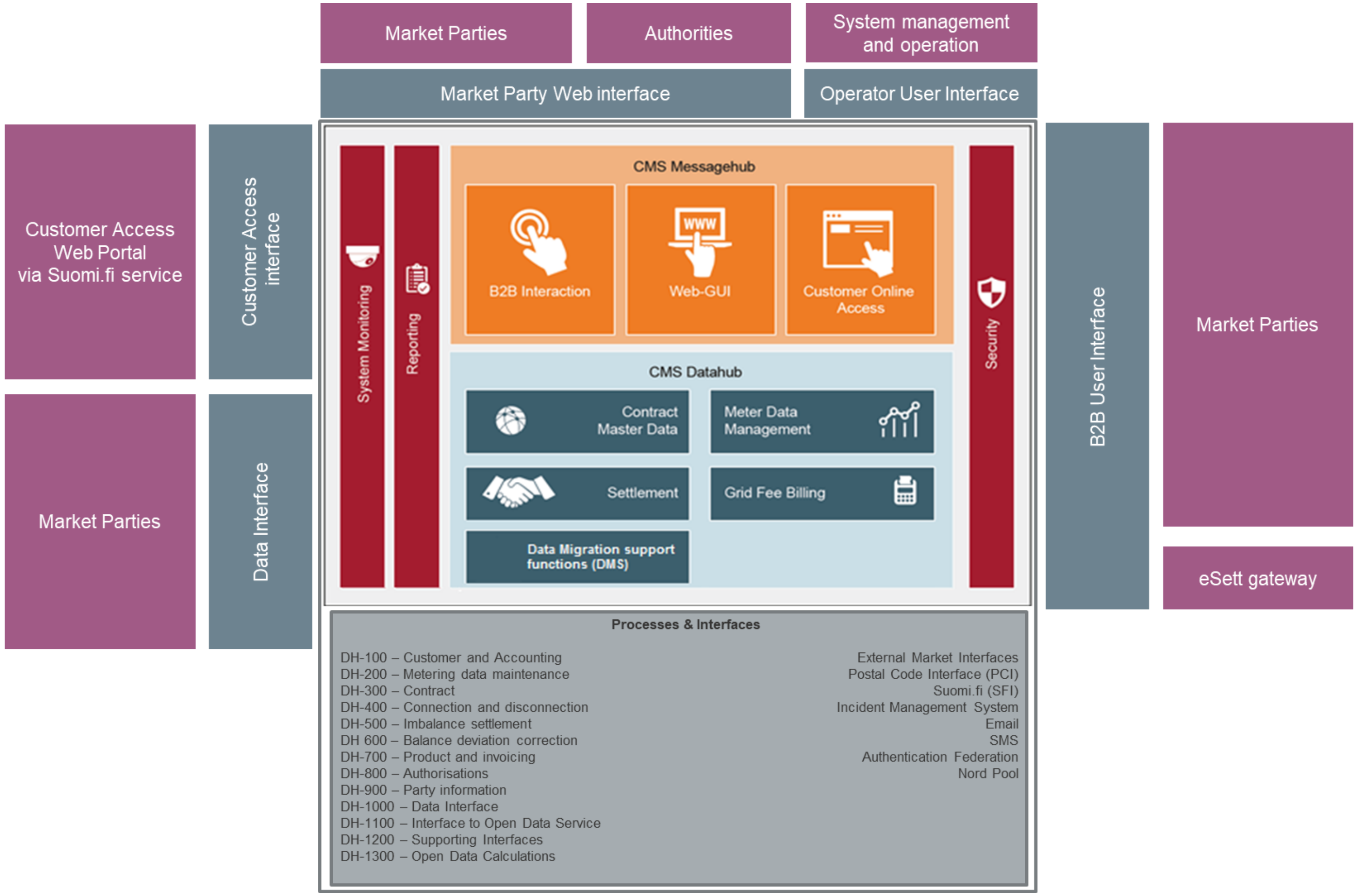
The block in the middle represents AgileDX (CMS) with its modules.

The surrounding grey/blue boxes are the primary interfaces exposed to market participants:

- Market Party Web Interface – maps on the Web-GUI of CMS, used by all market parties including the market operator.
- Operator User Interface – maps on the Web-GUI of CMS
- Customer Access Interface – maps on the Customer Online Access of CMS
- Data Interface – maps on the B2B interaction of CMS
- B2B User Interface – maps on the B2B interaction of CMS

The purple boxes identify the type of users interacting with the system.

The grey box at the bottom identifies the business processes and other interfaces supported by the solution.



Market Parties
Authorities
System management and operation
Market Party Web interface
Operator User Interface
Customer Access Web Portal via Suomi.fi service
Customer Access interface
Market Parties
Data Interface
System Monitoring
Reporting
CMS Messagehub
B2B Interaction
Web-GUI
Customer Online Access
CMS Datahub
Contract Master Data
Meter Data Management
Settlement
Grid Fee Billing
Data Migration support functions (DMS)
Security
B2B User Interface
Market Parties
eSett gateway

Processes & Interfaces

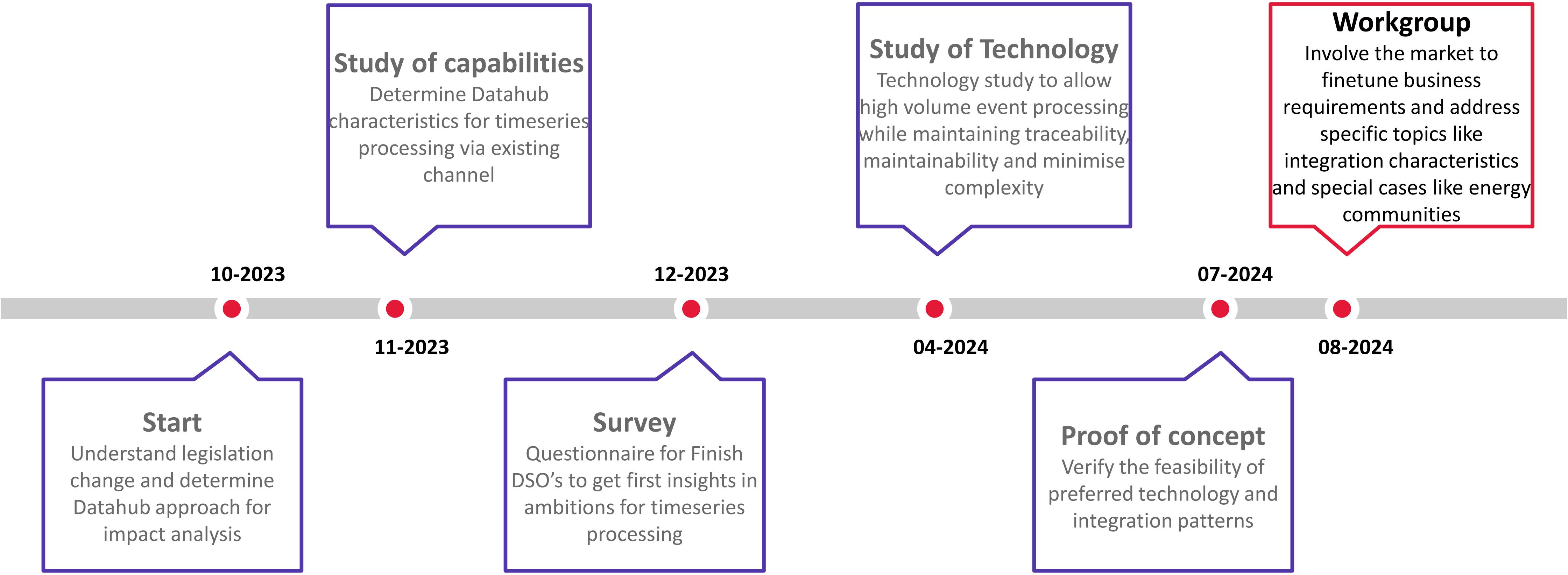
DH-100 – Customer and Accounting
DH-200 – Metering data maintenance
DH-300 – Contract
DH-400 – Connection and disconnection
DH-500 – Imbalance settlement
DH 600 – Balance deviation correction
DH-700 – Product and invoicing
DH-800 – Authorisations
DH-900 – Party information
DH-1000 – Data Interface
DH-1100 – Interface to Open Data Service
DH-1200 – Supporting Interfaces
DH-1300 – Open Data Calculations

External Market Interfaces
Postal Code Interface (PCI)
Suomi.fi (SFI)
Incident Management System
Email
SMS
Authentication Federation
Nord Pool

Background

Timeline

Background



10-2023

Start
Understand legislation change and determine Datahub approach for impact analysis

11-2023

Study of capabilities
Determine Datahub characteristics for timeseries processing via existing channel

12-2023

Survey
Questionnaire for Finish DSO’s to get first insights in ambitions for timeseries processing

04-2024

Study of Technology
Technology study to allow high volume event processing while maintaining traceability, maintainability and minimise complexity

07-2024

Proof of concept
Verify the feasibility of preferred technology and integration patterns

08-2024

Workgroup
Involve the market to finetune business requirements and address specific topics like integration characteristics and special cases like energy communities

Legislation

Background

The decree 767/2021 “Valtioneuvoston asetus sähköntoimitusten selvityksestä ja mittauksesta” (Government Decree on the Settlement and Measurement of Electricity Supplies) chapter 6, § 5 states

“network operator's information system processing metering data shall collect the registered measurement data from the new remote metering equipment into the metering data reading system at least every six hours”.

According to the Finnish regulator the purpose of the requirement is **to provide measurement data to the end consumer at least within six hours**. As **datahub** can be appointed as point of delivery for measurement data the decree requirement also applies to the datahub system.

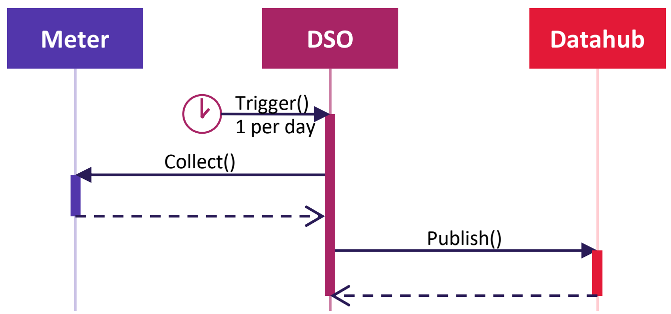
The requirement is limited to the **next generation** of smart meters and comes into force **1.1.2026**.

NextGen Meters

Background

GEN I

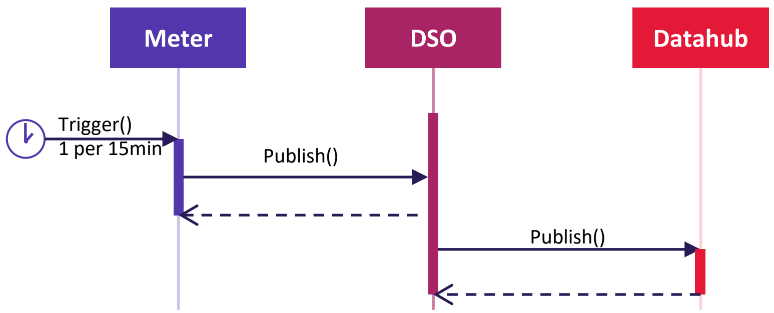
First Generation meters only provide APIs to request for measurement data and do not have the capability to push measurement data.



A batch driven approach is more suitable by collecting measurement for a period, validate and then publish for other consumers

GEN II

Second Generation meters have the capability to push the available measurement data as it becomes available.



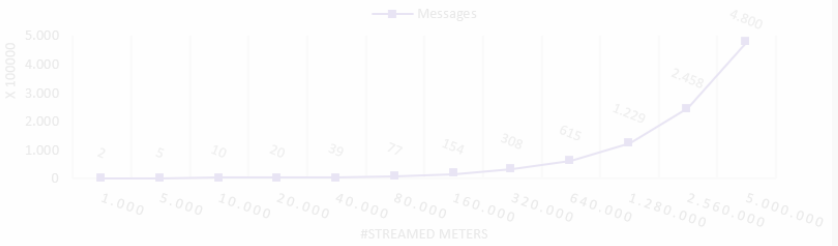
This allows for a more event driven approach to the collection, validation a publication of the available measurement data

Datahub characteristics – Study of Capabilities

Background

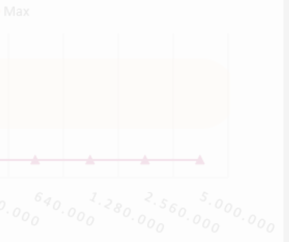
Meter to message ratio

- Approximately 75.000 E66 messages a day
- Containing 6-7 million transactions.
- Average batch size over the day is 80-100 transactions per message.
- Assuming the introduction of timeseries being send to the datahub individually the amount of timeseries messages a day will increase rapidly up to a theoretical maximum of **~480 mio a day** when all meters will stream their individual values.



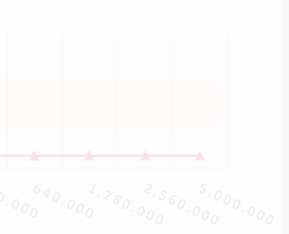
Message hub

- Currently on c Messagehub processes ~3.6 mio messages a day.
- On peak c
- Minimal in indicates c
- Message I impacted
- Roughly a to keep up



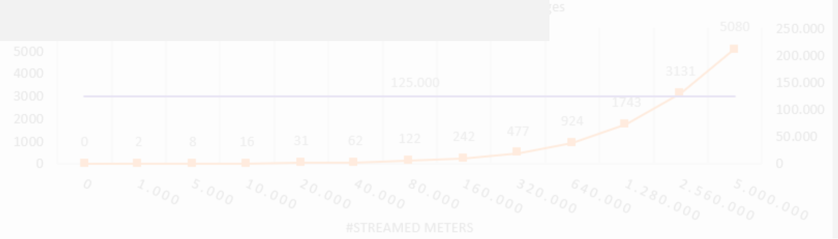
MDM

- Currently
- MDM is nc
- MDM inclu processed
- MDM is ex
- Roughly a manual in
- Distributio amount of



Bundling

- Bundling the information into batch messages will prevent hit limits in both Messagehub but more importantly in MDM
- Assuming the limit for MDM is around 250k messages a day and assuming values for an individual 15 min interval are communicated then **bundling of up to ~5.000** accounting points in a single message is required to stay well below the limit (**~125k messages per day**).



✓

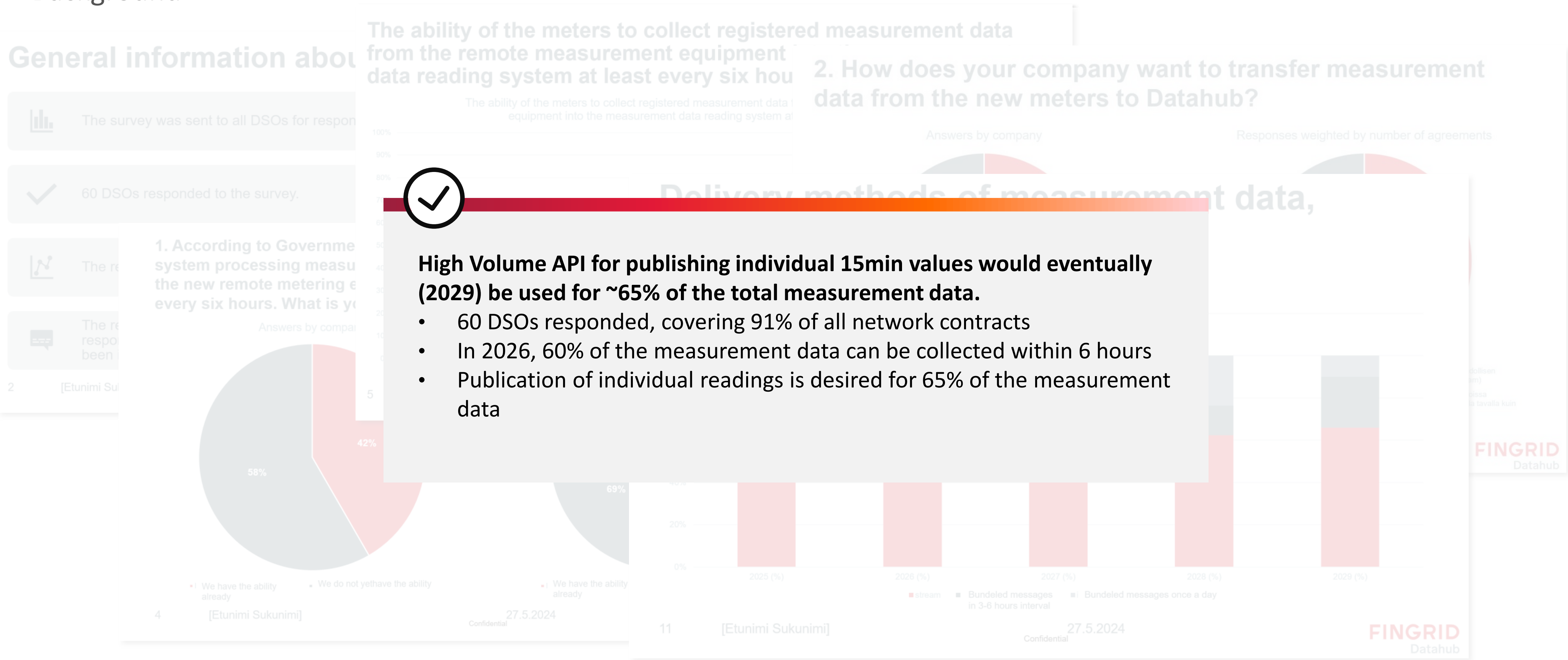
The existing channel for receiving timeseries data can be categorized as a batch interface. Designed for current Fingrid situation with additional validations optimized for current integration characteristics.

As a result the current interface is expected to be able to process up to ~250k – 500k messages per day.

Currently Datahub processes around 75k messages a day which in total contains around 500 mio timeseries

Fingrid Survey

Background

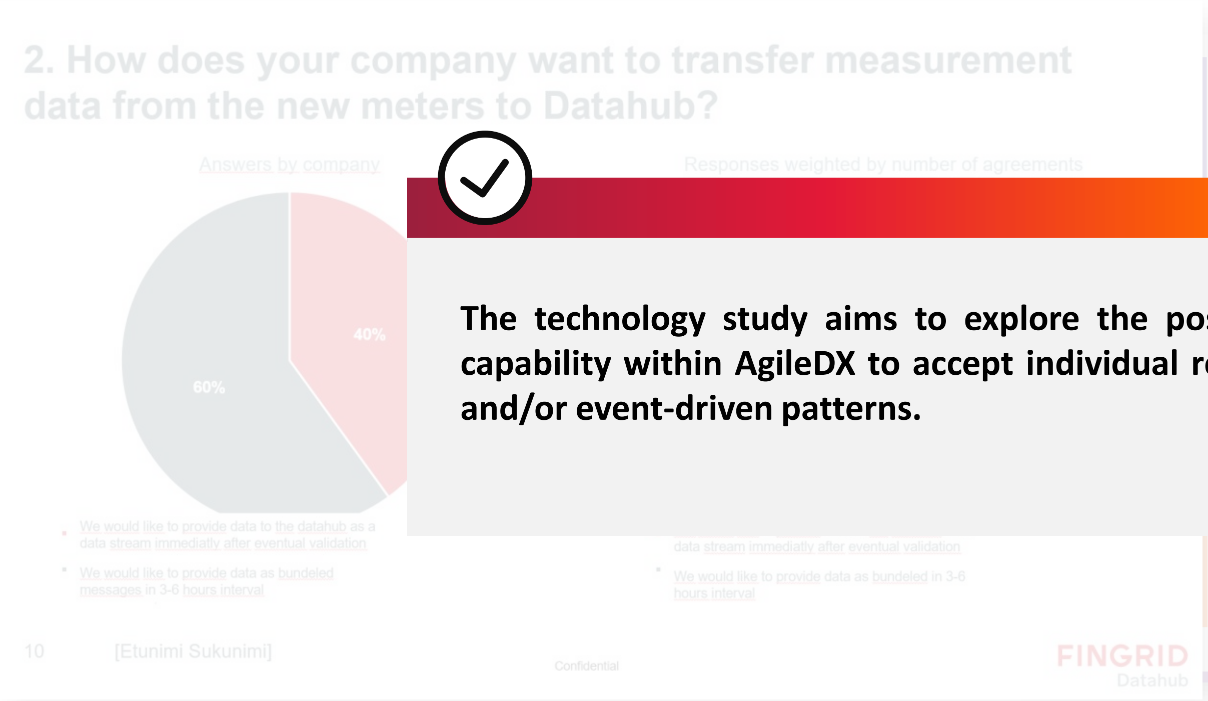

General information abou
The survey was sent to all DSOs for respon
60 DSOs responded to the survey.
1. According to Governme system processing measu the new remote metering e every six hours. What is yo
The ability of the meters to collect registered measurement data from the remote measurement equipment data reading system at least every six hou
2. How does your company want to transfer measurement data from the new meters to Datahub?
Answers by company
Responses weighted by number of agreements
Delivery methods of measurement data,
58%
42%
69%
We have the ability already
We do not yethave the ability
Bundeled messages in 3-6 hours interval
Bundeled messages once a day
2025 (%)
2026 (%)
2027 (%)
2028 (%)
2029 (%)
[Etunimi Sukunimi]
27.5.2024
Confidential
FINGRID Datahub

High Volume API for publishing individual 15min values would eventually (2029) be used for ~65% of the total measurement data.

- 60 DSOs responded, covering 91% of all network contracts
- In 2026, 60% of the measurement data can be collected within 6 hours
- Publication of individual readings is desired for 65% of the measurement data

Focus Area Technology Study

Background



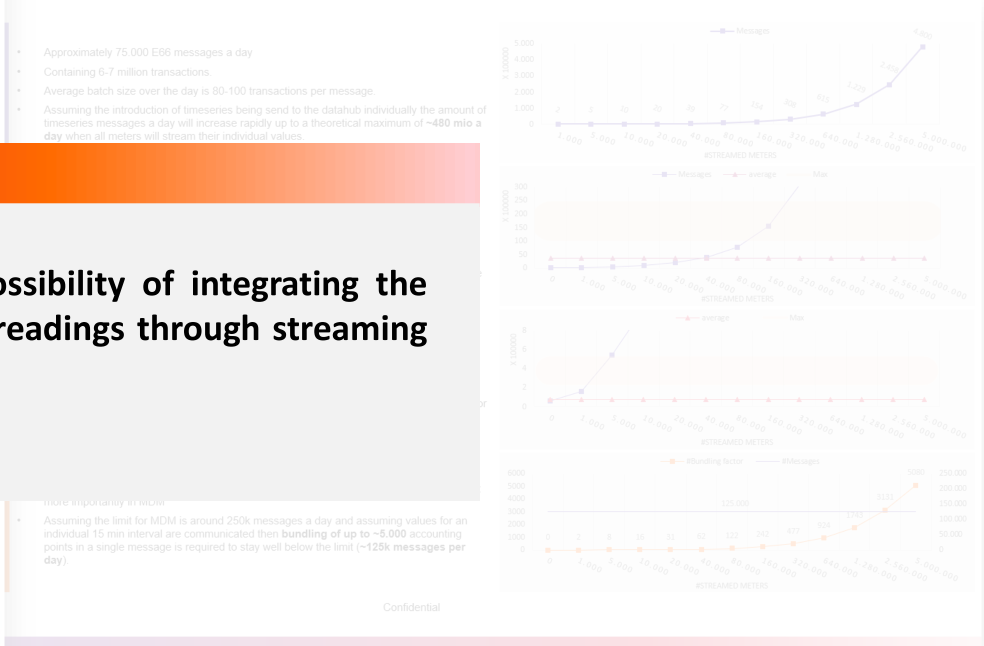
2. How does your company want to transfer measurement data from the new meters to Datahub?

- We would like to provide data to the datahub as a data stream immediatly after eventual validation
- We would like to provide data as bundeled messages in 3-6 hours interval

- data stream immediatly after eventual validation
- We would like to provide data as bundeled in 3-6 hours interval

10 [Etunimi Sukunimi] Confidential

FINGRID Datahub



- Approximately 75.000 E66 messages a day
- Containing 6-7 million transactions.
- Average batch size over the day is 80-100 transactions per message.
- Assuming the introduction of timeseries being send to the datahub individually the amount of timeseries messages a day will increase rapidly up to a theoretical maximum of **~480 mio a day** when all meters will stream their individual values.

more importantly in MDM

- Assuming the limit for MDM is around 250k messages a day and assuming values for an individual 15 min interval are communicated then **bundling of up to ~5.000** accounting points in a single message is required to stay well below the limit (**~125k messages per day**).

Confidential

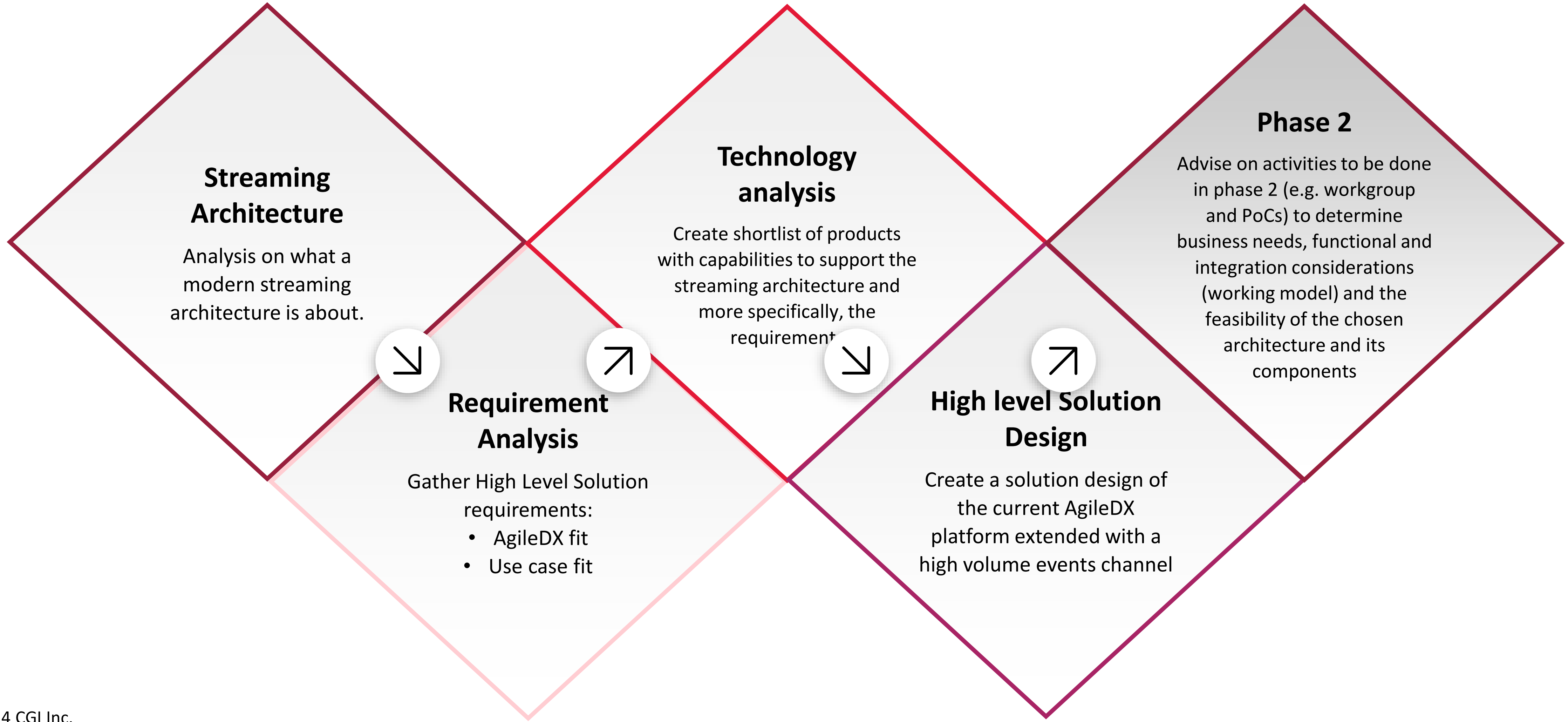
✓

The technology study aims to explore the possibility of integrating the capability within AgileDX to accept individual readings through streaming and/or event-driven patterns.

Technology Study

Approach

Technology Study



Streaming Architecture

Analysis on what a modern streaming architecture is about.

Requirement Analysis

Gather High Level Solution requirements:

- AgileDX fit
- Use case fit

Technology analysis

Create shortlist of products with capabilities to support the streaming architecture and more specifically, the requirement

High level Solution Design

Create a solution design of the current AgileDX platform extended with a high volume events channel

Phase 2

Advise on activities to be done in phase 2 (e.g. workgroup and PoCs) to determine business needs, functional and integration considerations (working model) and the feasibility of the chosen architecture and its components

What is streaming data

Technology Study

Streaming data is data that is emitted at high volume in a continuous, incremental manner with the goal of low-latency processing.

Characteristics

✓ **Chronologically significant**
Individual elements in a data stream contain time stamps

✓ **Continuously flowing**
A data stream has no beginning or end

✓ **Unique**
Repeat transmission of a data stream is challenging because of time sensitivity

✓ **Nonhomogeneous**
Some sources may stream data in multiple formats that are in structured formats

✓ **Imperfect**
Temporary errors at the source may result in damaged or missing elements in the streamed data

Source: *©Amazon - https://aws.amazon.com/what-is/streaming-data/*

Modern Data Streaming Architecture

Technology Study

Stream Sources

Your source of streaming data includes data sources like sensors, social media, IoT devices, log files generated by using your web and mobile applications, mobile devices that generates semi-structured and unstructured data as continuous streams at high velocity.

- Smart meters
- Database Change
- Sensor networks
- Social Media feeds

Stream Ingestion

The stream ingestion layer is responsible for **ingesting data into the stream storage layer**. It provides the ability to collect data from tens of thousands of data sources and ingest in **near real-time**.

- HTTP/2
- WebSocket
- MQTT **(Message Queuing Telemetry Transport)**
- AMQP **(Advanced Message Queuing Protocol)**

Stream Storage

The stream storage layer is responsible for providing **scalable** and **cost-effective** components to store streaming data. The streaming data can be **stored in the order it was received** for a **set duration of time** and can be **replayed indefinitely during that time**.

- Kafka
- RabbitMQ

Stream Processing

The stream processing layer is responsible for **transforming data into a consumable state** through data validation, cleanup, normalization, transformation, and enrichment. The streaming records are read in the order they are produced, allowing for real-time analytics, building event driven applications, or streaming ETL.

- Kafka Streams
- Spark
- Flink
- Storm

Stream Destination

The destination layer is like a purpose-built destination depending upon your use case. Your destination can be an event driven application, data lake, data warehouse, database, or an OpenSearch.

- Database (e.g MongoDB)
- Storage (e.g E3)
- Etc.

Source: *©Amazon - https://docs.aws.amazon.com/whitepapers/latest/build-modern-data-streaming-analytics-architectures/what-is-a-modern-streaming-data-architecture.html*

Main Requirements per Solution area

Technology Study

Stream Sources

Stream Ingestion

1. **Minimize complexity** for event producers to send their events by utilizing open standards and common interaction patterns;
2. Primarily support for up to **40.000.000 events per 15min**;
3. Authentication and authorization relying on **existing IAM implementation** (AD, certificates, partners);
4. **Minimize impact on existing platform** (from maintenance, development and deployment perspective), reuse existing components where possible.

Stream Storage

1. Primarily support for up to **1.000.000.000 events per day** and technically scalable up to 5.000.000.000 events per day
2. **Open Source where possible** (no additional licenses); Third party (contracting)dependencies kept to the minimum to allow for flexible scaling, deployment and predictable costing.
3. **Stream to (micro) Batch capability** to integrate with MDM module;
4. **Minimize impact on existing platform** (from maintenance, development and deployment perspective), reuse existing components where possible.

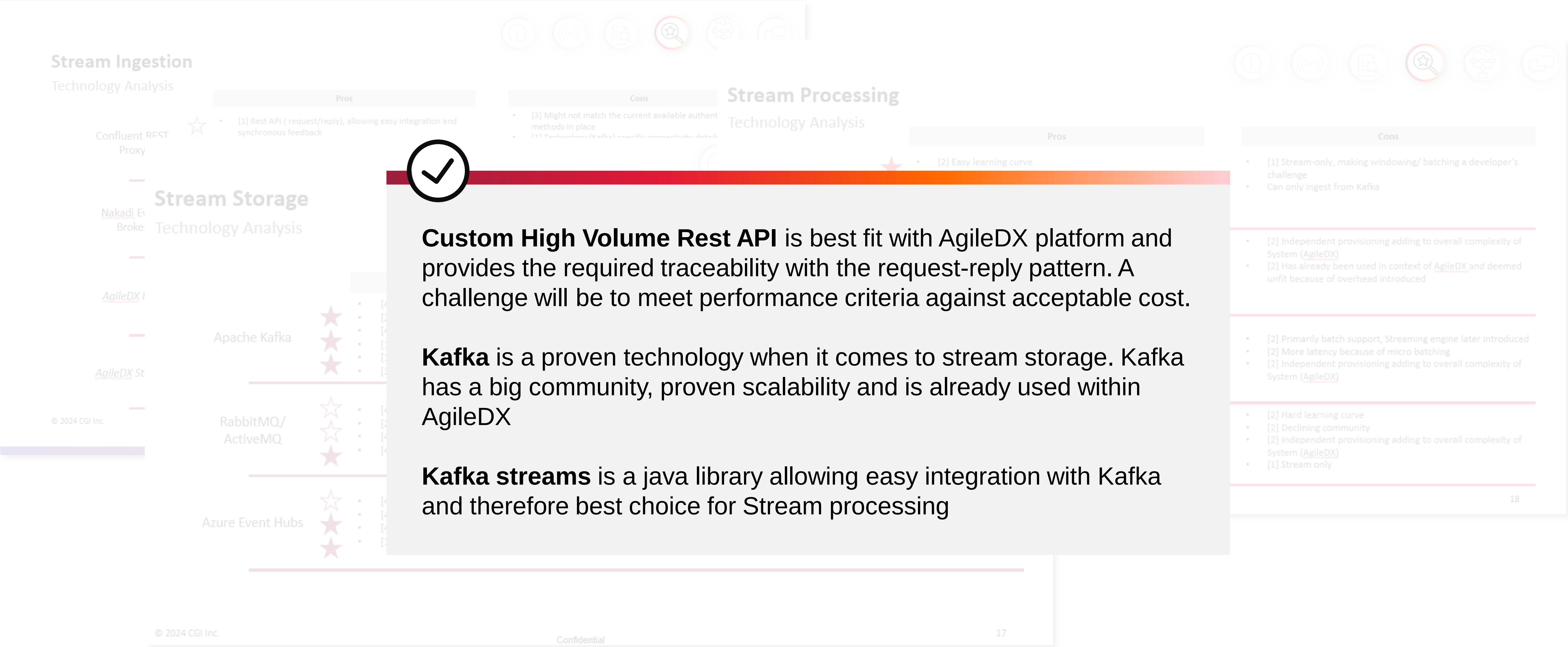
Stream Processing

1. **Stream to (micro) Batch capability** to integrate with MDM module;
2. **Minimize impact on existing platform** (from maintenance, development and deployment perspective), reuse existing components where possible.
3. **Open Source where possible** (no additional licenses); Third party (contracting)dependencies kept to the minimum to allow for flexible scaling, deployment and predictable costing.

Stream Destination

Technology Analysis

Technology Study



Stream Ingestion

Technology Analysis

Pros

Cons

Confluent REST Proxy

- [1] Rest API (request/reply), allowing easy integration and synchronous feedback

- [3] Might not match the current available authent methods in place

Nakadi E Broke

AgileDX I

AgileDX St

© 2024 CGI Inc.

Stream Storage

Technology Analysis

Apache Kafka

RabbitMQ/ ActiveMQ

Azure Event Hubs

© 2024 CGI Inc.

Confidential

17

Stream Processing

Technology Analysis

Pros

Cons

- [2] Easy learning curve

- [1] Stream-only, making windowing/ batching a developer's challenge
- Can only ingest from Kafka

- [2] Independent provisioning adding to overall complexity of System (AgileDX)
- [2] Has already been used in context of AgileDX and deemed unfit because of overhead introduced

- [2] Primarily batch support, Streaming engine later introduced
- [2] More latency because of micro batching
- [2] Independent provisioning adding to overall complexity of System (AgileDX)

- [2] Hard learning curve
- [2] Declining community
- [2] Independent provisioning adding to overall complexity of System (AgileDX)
- [1] Stream only

18

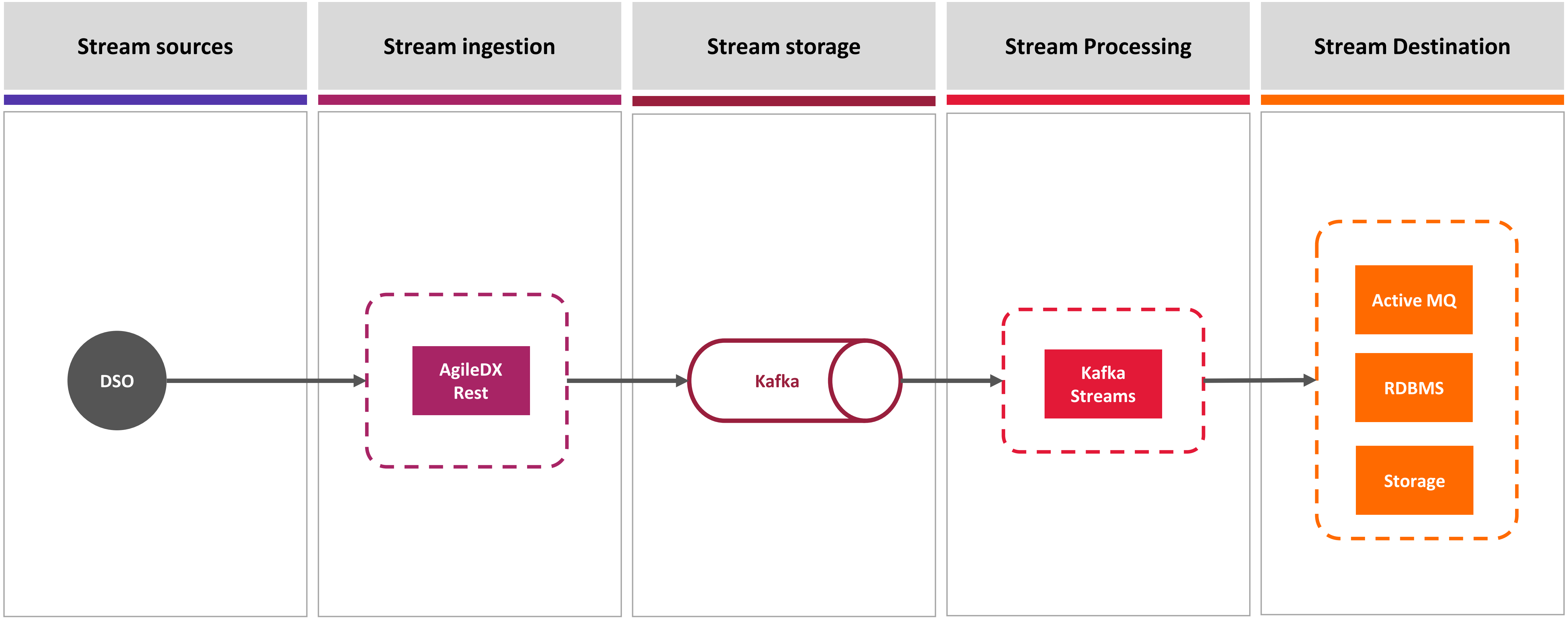
Custom High Volume Rest API is best fit with AgileDX platform and provides the required traceability with the request-reply pattern. A challenge will be to meet performance criteria against acceptable cost.

Kafka is a proven technology when it comes to stream storage. Kafka has a big community, proven scalability and is already used within AgileDX

Kafka streams is a java library allowing easy integration with Kafka and therefore best choice for Stream processing

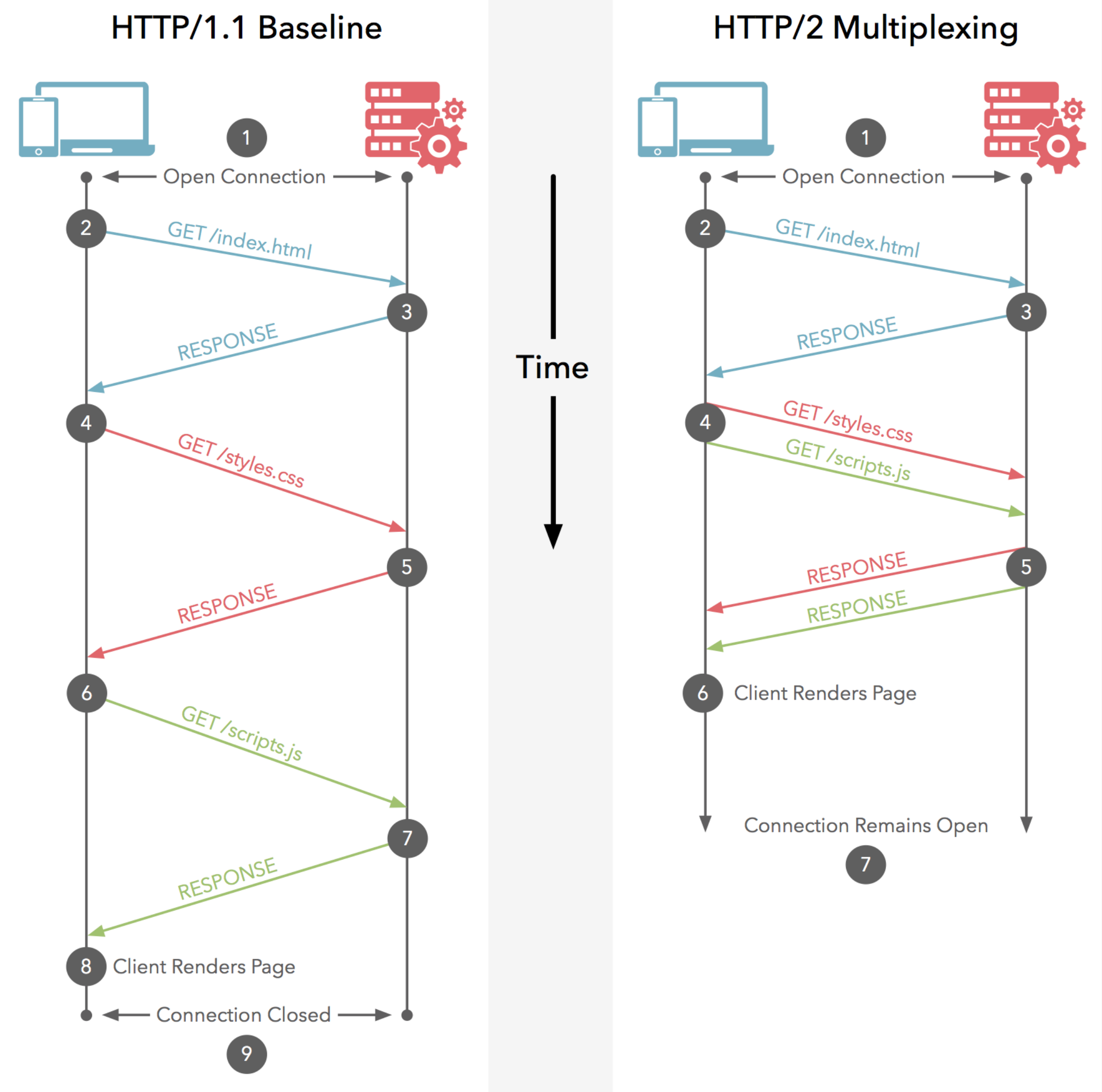
High Level Solution Design

Technology Study



HTTP/2

Technology Study



Proof of Concept

Phase 2

Proof of Concept

Stream ingestion

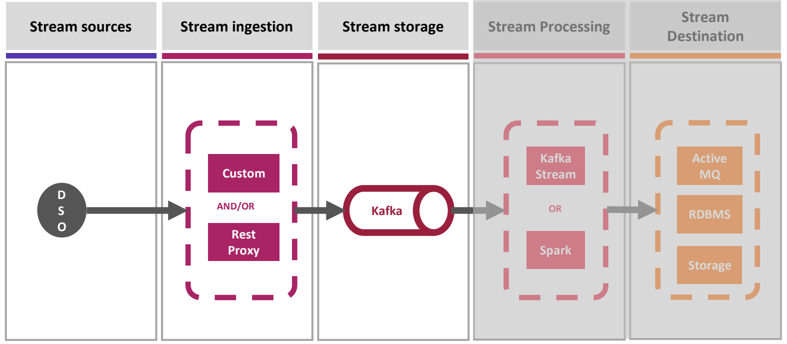
Proof of Concept

High performant ingestion channel integrated with available AgileDX logging and authentication modules. Verify pattern to be used in actual implementation

- [preferred] **REST API based on HTTP/2** preferred because of reduced complexity for participants and synchronous reply if a message has successfully been received.
- [Fallback] Stream API based on protocols like websocket, MQTT

Goal of the proof of concept is to confirm solution direction for receiving events with special attention to:

- Scalability (> 40 mio events per 15 minutes)
- Security integration
- Kafka architecture



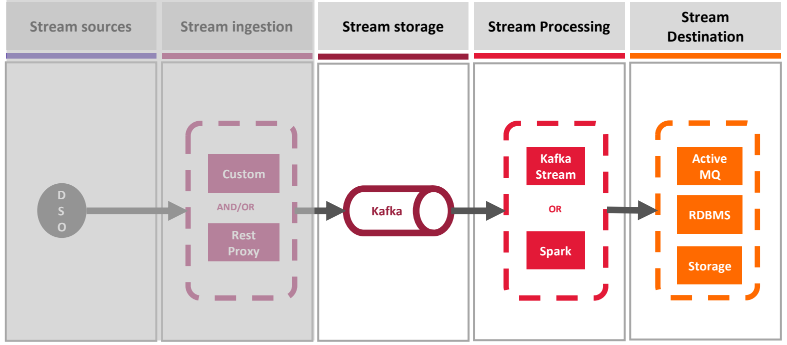
Stream Processing

Proof of Concept

Verify if **Kafka Streams** suffice as stream processor or additional technologies are required. If so, extend proof of concept with Apache Spark.

Goal of the proof of concept is to verify if Kafka Streams is able to process events and combine them to batch messages for MDM to process. Special attention goes to:

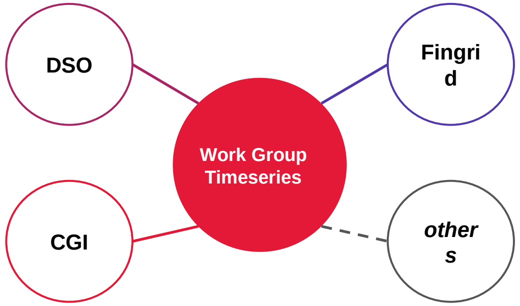
- Scalability (> 40 mio events per 15 minutes)
- Windowing capabilities
- Metadata requirements/constraints to allow for efficient Stream to Batch
- Baseline Kafka cluster configuration to allow Stream to Batch



Work group

Introduction of a high volume event handling channel will introduce technical and possibly functional deviations from the current way of working. To allow optimal fit for the Finnish market it is advised to have a work group organized with experts from Fingrid, DSO’s, Software vendors and CGI to agree on aspects like:

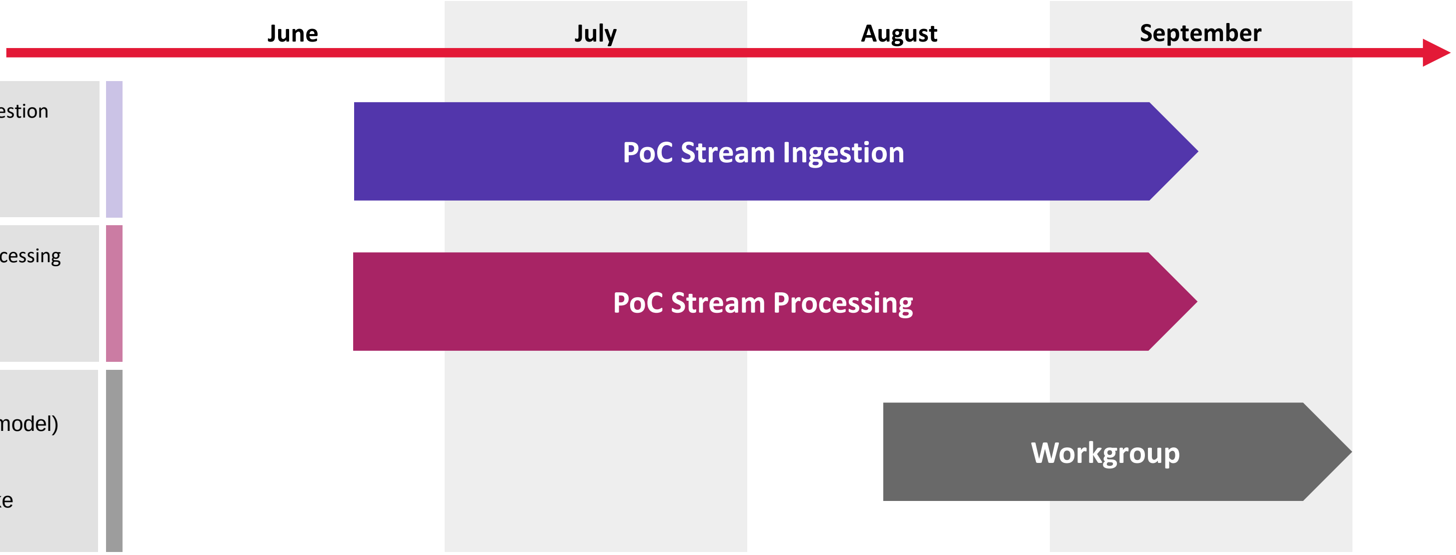
- Event format (attributes, identifiers)
- Patterns (Request/ Reply; Put and Update)
- Rejections and/or confirmations
- Distribution to other market parties.
- How to support specific scenarios like “energy communities”



Planning

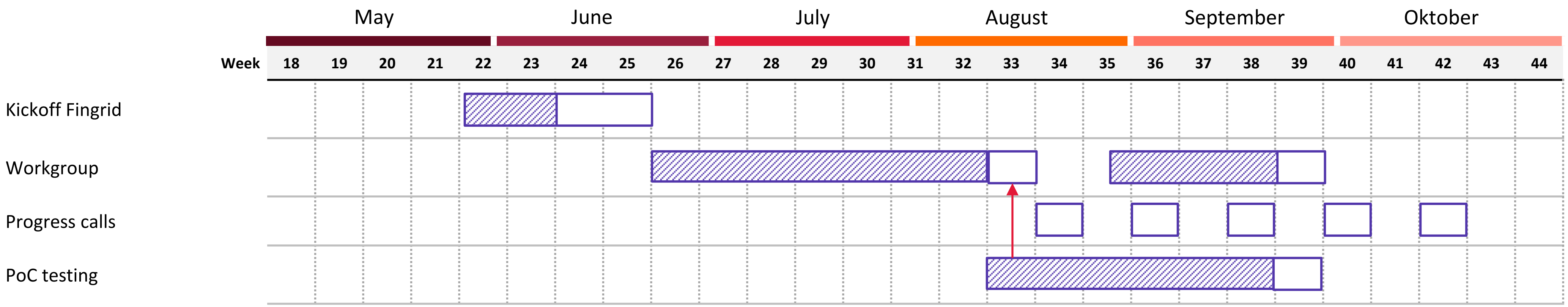
Proof of Concept

PoC Stream Ingestion	• Design/build/test/provision Stream ingestion • Loadtest Stream Ingestion • Optimize • Report
PoC Stream Processing	• Design/build/test/provision Stream processing • Loadtest Stream Processing • Optimize • Report
Workgroup	• Event format (attributes, identifiers) • Patterns (Request/ Reply; working model) • Rejections and/or confirmations • Distribution to other market parties. • How to support specific scenarios like “energy communities”



Workgroup Planning

Proof of Concept



- Kick-off fingrid to align on previous steps (phase 1 – report), future steps (phase 2), Preparation for workshop
- Workshop with sector.
  - 1st (13-14 August): Kickoff, Event format, Pattern, Rejection/Confirmation, status PoC
  - 2nd (25th September): Lookback, Distribution to other market parties, Special cases, Testing days
- Progress calls: Biweekly digital call to address progress on actions from workshop (only when needed)
- PoC Testing: 2 days (23-24 September) onsite for connecting selected parties to sandbox. During 1st workshop the Test tooling as used by CGI will be made available to stakeholders.

□ Week meeting will be held

▨ Lead/preparation time for participants. Supported with ad-hoc meetings if needed

Phase 2

Proof of Concept

PoC

CGI is responsible for the deliverable
Fingrid is informed

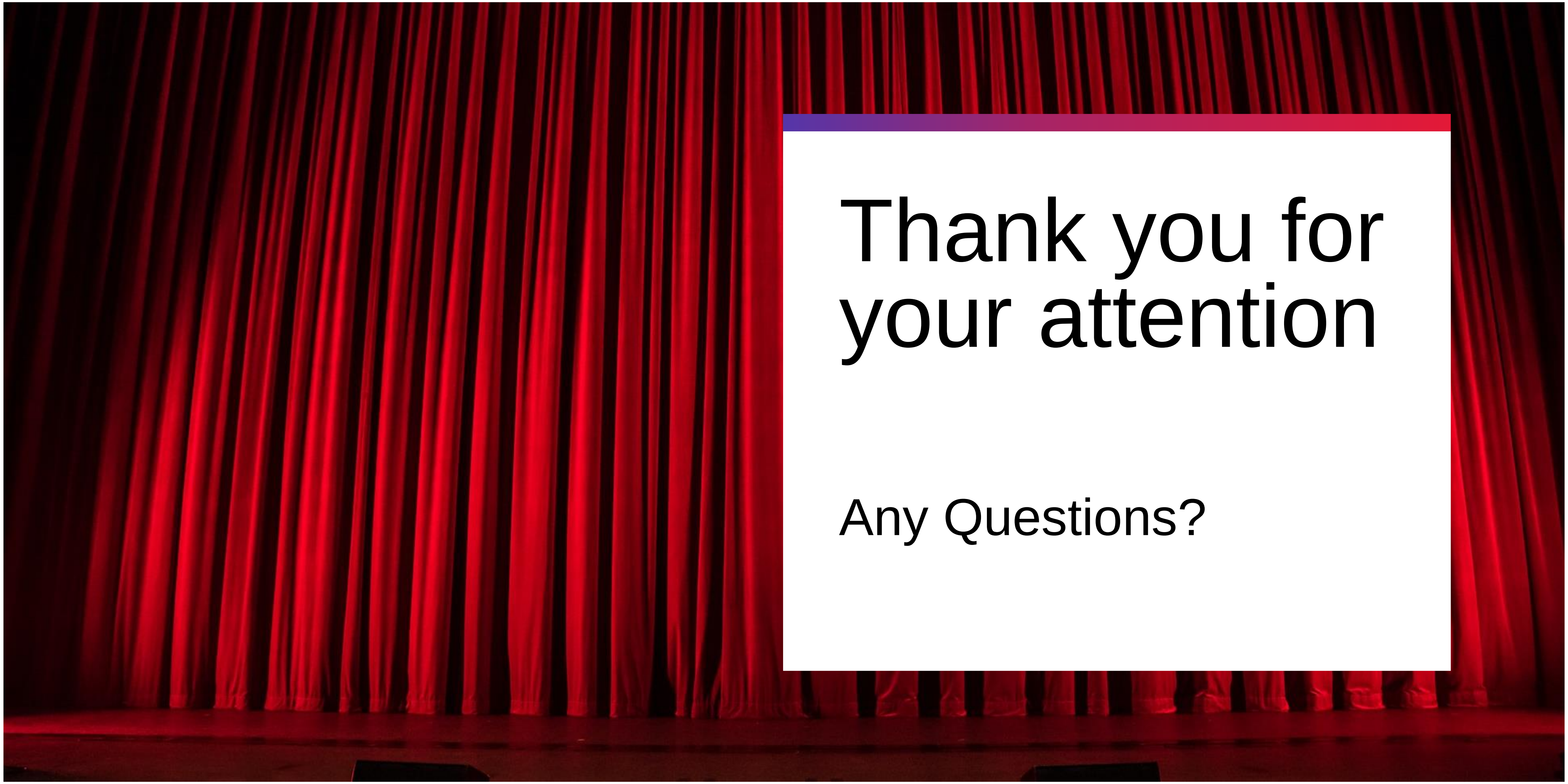
- ✓ **Report - Event Channel Timeseries** describing the results of the proof of concepts including the feasibility of the technology and a continuation plan in case technology is deemed unfeasible
- ✓ **High Level Design - Event Channel Timeseries** including design considerations, decisions and dependencies and expected on premise resource requirements, to be used as input for System Design/Architecture
- ✓ **Test Tool** (driver written in NodeJS) shared with workgroup, prior to testing days, as reference.

Workgroup

CGI (and Fingrid) responsible for the deliverable
Workgroup is consulted

- ✓ **Technical Interface Specification** where Fingrid and CGI are responsible for creating the draft and workgroup is consulted
- ✓ Memo per main topic addressed by workgroup
  - **Integration pattern** (Request/Response/ working model)
  - **Traceability** (confirmation/ rejection)
  - **Special cases** (e.g. Energy Communities)
  - **Distribution of Timeseries** to DSO and Suppliers
  - Concept **Test Planning** and environment dependencies phase 3

  These memos are to be used as input for System Design/Architecture



Thank you for your attention

Any Questions?