

# Common Statistical Production Architecture (CSPA): Restating Sharing Made Easy

Workshop on Statistical Production Architecture and Software Sharing

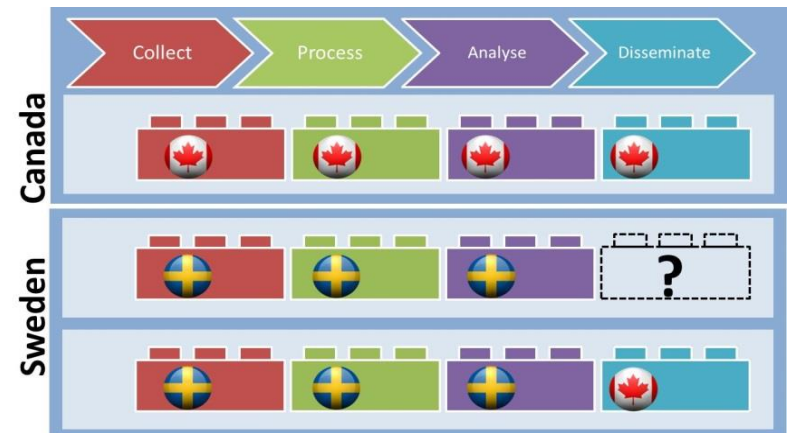
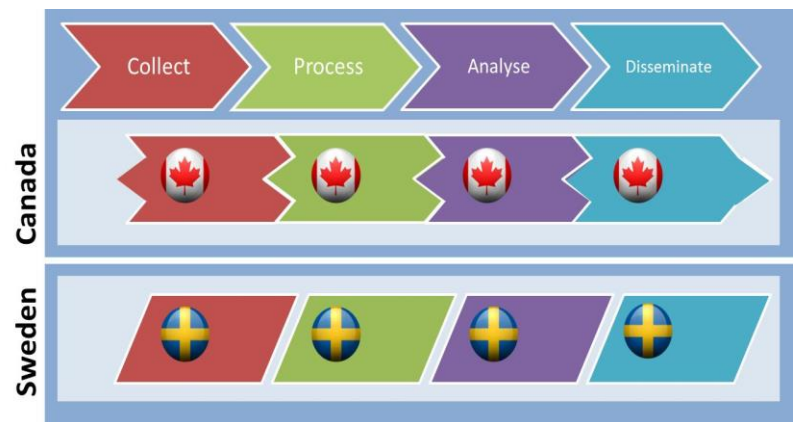
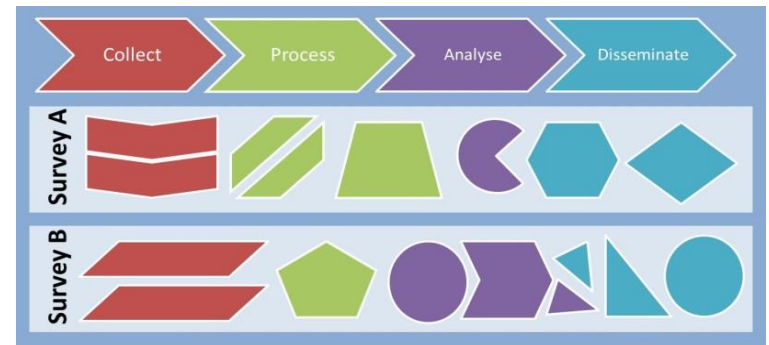
*5-6 December 2018, Belgrade, Serbia*

# Common Statistical Production Architecture (CSPA)

- Template architecture for the statistical community (across GSBPM processes)
- Includes application and technology architecture and principles (service oriented)
- Set of standard specifications for new statistical components (services) that can be used in a modular way
- Does not prescribe technology environments of statistical organizations
- A new way of developing statistical tools, with shareability as a design feature, not an afterthought
- Framework for collaborating and sharing




# CSPA Concept

1. Specialized business processes & IT solutions
  2. Enterprise Architecture
  3. Common Statistical Production Architecture
- But:
    - Not plug-and-play
    - Few CSPA compliant services developed past 5 years



# What we Learned



- We consulted and engaged with (potential) users at 2017 Wiesbaden workshop and 2018 ModernStats World Workshop
- It made us realize that:
  - CSPA ‘compliance’ was perceived as a barrier 
  - The CSPA Document did not invite to commence with implementation or sharing of services 
  - CSPA Catalogue was not easy to find and to access
  - The work on application architecture patterns (adapters and containerization) simplified the understanding of CSPA as a sharing concept 

# So we

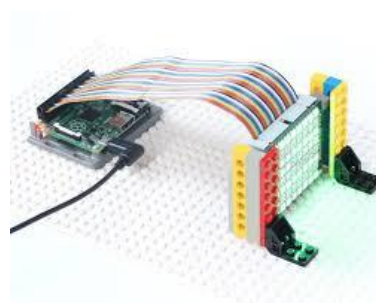
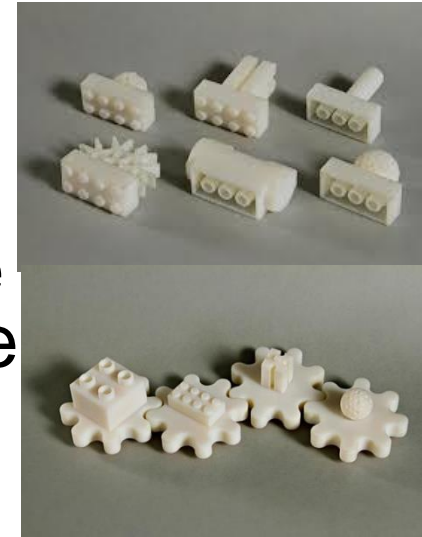
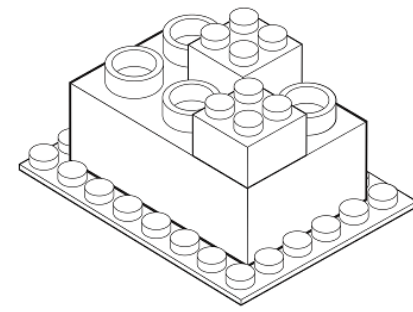


- Listened to and reflected on users' concerns
- Repositioned and restated CSPA as first and foremost: a concept to encourage sharing
- Defined Requirements and Features as the key to increasing shareability
- Improved the CSPA Service Catalogue and increased the availability of services and their documentation (Eurostat&ESSnet)



# CSPA Adapters

- Small piece of functionality that communicates with the core business-logic of the service, and with the outside world
- Technical “glue” for binding the CSPA Service to a local NSI’s environment while still retaining the core business logic of the Service
- Platforms flexible: world changing around us







# Goals of CSPA

- The CSPA was developed to support the sharing and re-use of tools across statistical domains and between statistical organisations.
- It provides a blueprint for a new way of designing, building and implementing the tools needed to produce official statistics.
- We're still trying to figure out how CSPA can guide statistical organisations to build services that can be shared.





# What is CSPA?

What CSPA is not:

- Prescriptive set of rules to be met to be compliant
- Plug-and-Play (*but could be*)

But Neither:

- Free format
- Umbrella that covers any statistical service

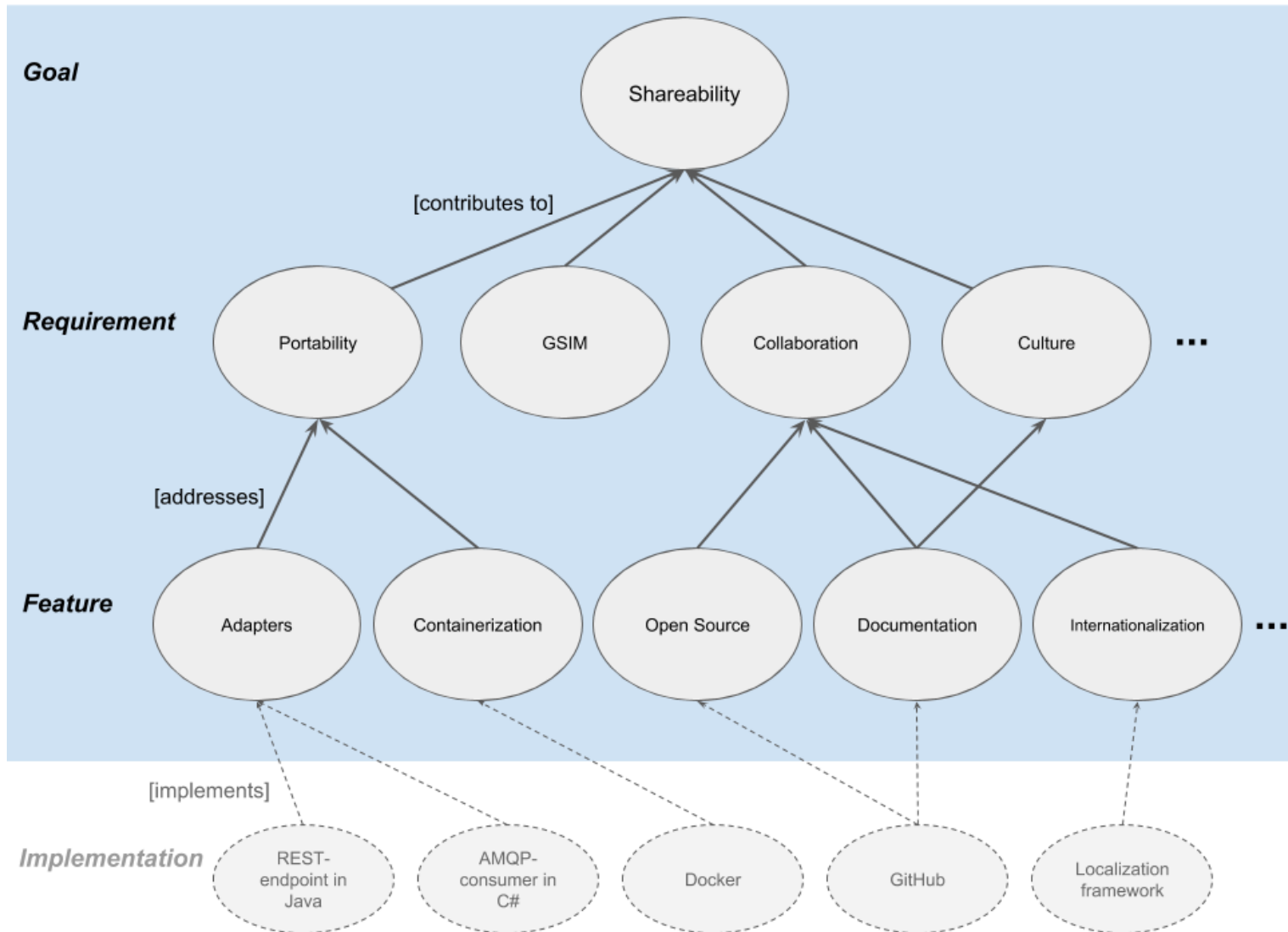
It is list of principles and features that make sharing easier:

- The more met, the easier to share, they higher CSPAness, CSPA compatability, CSPA shareability

# Application architecture

- Describes the behavior of applications and services used in a business
- Application architecture helps you with how the different applications, components and services works together
- CSPA relies on existing application architecture patterns, and best practices
- We have curated and contextualized different patterns and best practices

# From goal to implementation





CONTEXT IS KING



# Context

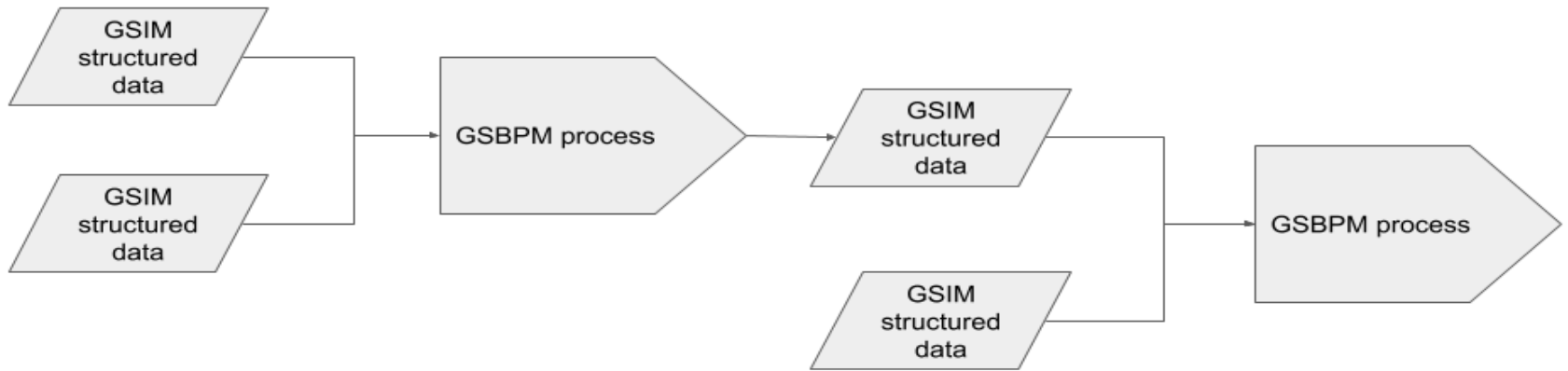


- The statistical services do not serve any kind of process, they should serve **GSBPM** processes.
- This means that services should be able to process **GSIM-structured data** rather than just any data.
- This means that services do not have to support any type of organization, but a **statistical organization**
- Make use of existing technology investments as much as possible
- Build on the shoulders of giants and use our internal experts to work on the problems that are **specific to statistics**

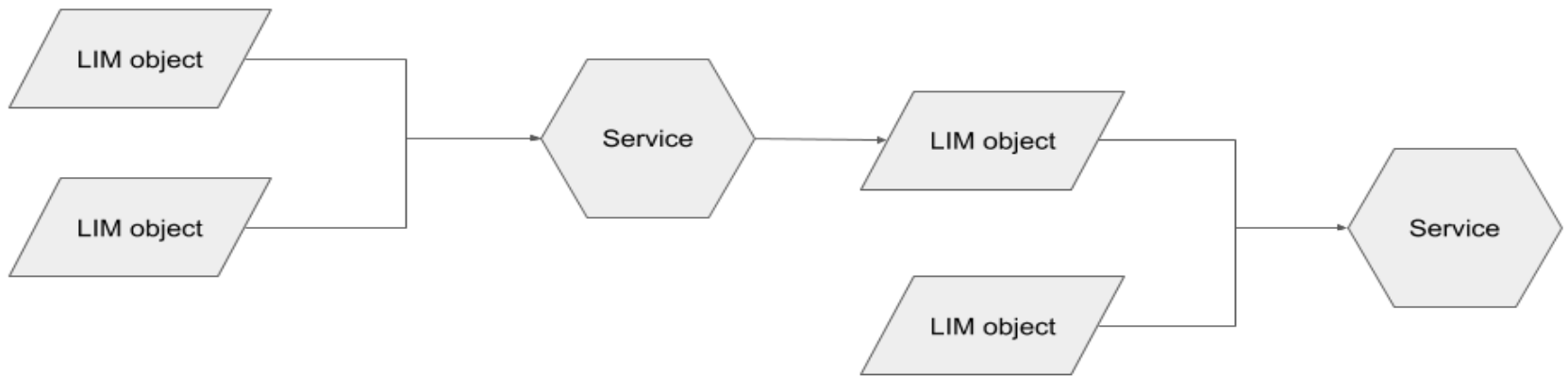
# CSPA Requirements

- R1 Collaboration: replicate, expand & extend
- R2 GSIM Alignment: logical/semantic
- R3 GSBPM Alignment: serve GSBPM processes
- R4 Metadata Driven: input/output configuration / parameters/ design
- R5 Statistical Program alignment: domain context
- R6 Secure data management: identities, roles and data classification (traceability, accountability, confidentiality)
- R7 Integration to an existing organization: IT & training users
- R8 Proper governance: service/release management
- R9 Culture context: mindset/trust sharing
- R10 Performance: minimal impact on other services
- R11 Portability: between computing environments

# R4 Metadata driven



Conceptual process diagram.



Service implementation diagram.

# Features to implement requirements

- Services can have one or more features that implement one or more requirements
- Features are the things you need to add to your service to make it more shareable
- Features are more than technical components
- Examples can be Adapters, Documentation, using GSIM, containerization etc.

# CSPA Features

- F1 Documentation
- F2 Internationalization
- F3 Open Source
- F4 CSPA Adapters
- F5 Using GSIM: common language
- F6 Security solutions
- F7 Metadata access
- F8 Sandboxing for exploration
- F9 Containerization
- F10 Multiple instances
- F11 Versioning
- F12 Support for Centralized data
- F13 Support for Decentralized data
- F14 Service integration
- F15 Big Data capable
- F16 Virtualization
- F17 Context aware
- F18 Human interaction
- F19 Resilience



# Example

## CSPA and java-vtl

**java-vtl** (<https://github.com/statisticsnorway/java-vtl>) is an Open Source Java implementation of the [Validation Transformation Language](#), based on the VTL specification. The implementation follows the JSR-223 Java Scripting API and exposes a simple connector interface one can implement in order to integrate with any data stores. VTL is a standard language for defining validation and transformation rules (set of operators, their syntax and semantics) for any kind of statistical data. The core functionality of java-vtl is to provide a interpreter that can interact with data.

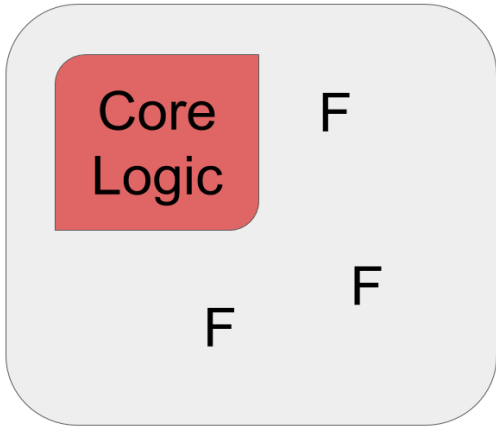
Feature	Comments
F1 Documentation	The java-vtl implementation provides an interactive documentation describing how to use VTL. ( <a href="https://statisticsnorway.github.io/java-vtl/reference/">https://statisticsnorway.github.io/java-vtl/reference/</a> )
F3 Open Source	<b>java-vtl</b> is on GitHub using a permissive Apache 2.0 license.
F4 CSPA Adapters	The connector interface provided with java-vtl makes it easy to implement support for more data sources.
F5 Using GSIM	The internal representation of data uses the part of GSIM that specifies Data structures.

# Algorithmia example

Feature	Comments
F1 Documentation	Algorithmia offers a standardized way to write documentation and encourages service builders to add documentation while building. We recommend to write all documentation in English.
F2 Internationalization	Algorithmia offers no way to interaction with the service. It is only algorithms, there is no GUI. Any qualifiers used when giving input or output are fixed, they cannot be changed depending on the user.  We recommend to use only English names (for algorithms, parameters, etc.)
F3 Open Source	Algorithmia forces service builders to choose either an open source license for their algorithms or to mark them as closed sources. One of the options is GNU GPL v3. It is not clear whether Algorithmia "acts" on this information, for example forcing an algorithm that uses another GPL3 algorithm to be open source as well.
F4 CSPA Adapters	Within Algorithmia, builders are very limited in creating dependencies. The platform abstracts from the underlying infrastructure for example. The main dependencies are of other services or data sources. Algorithmia does not enforce abstraction in terms of data format. You can send any data to an algorithm (from strings to JSON files to binary files).  We recommend that algorithms are created that can convert JSON data holding GSIM objects into standardized data structures that can be used within the algorithms.
F5 Using GSIM	As stated with F4, Algorithmia puts no constraints on the structure of inputs and outputs.  We recommend to agree on standardized JSON formats describing GSIM objects.
F6 Security solutions	All data within the Algorithmia platform is locally accessible by a specific algorithms. Communication with algorithms is encrypted (HTTPS), connections from algorithmia to external data sources are encrypted as well. Algorithmia has the option to confine algorithms in such a way that they cannot access the Internet. The Algorithmia platform itself runs on a professional cloud platform. Therefore, quite some security comes out of the box.
F7 Metadata	As stated in F4 and F5, Algorithmia is not statistics-aware. This can be done by introducing GSIM on all I/O operations.
F8 Sandboxing for exploration	Algorithmia comes by default with an environment that allows for live use of services, to experiment with input and see what the output will be.
F9 Containerization and F16	Algorithmia employs containerization and virtualization techniques to abstract from the underlying technology. It is not possible

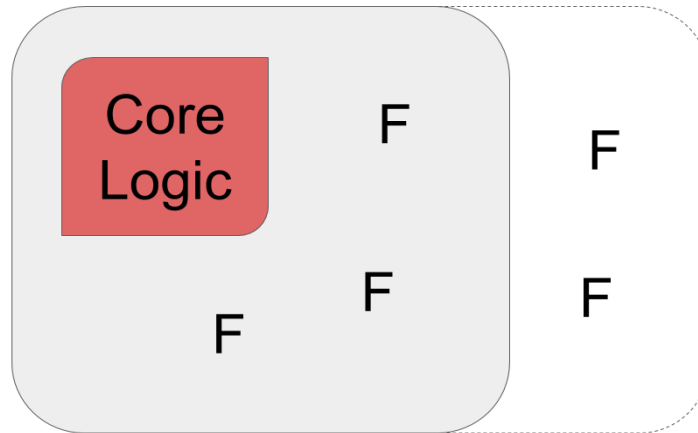
F10 Multiple instances	Algorithmia automatically scales up when there is peak use of any service. Of course, if there are references to external data, these external sources would have to scale up as well.
F18 Human interaction	Algorithmia does not offer human interaction with the service besides supplying input and consuming output. Services that require human interaction will have to be implemented on another platform.
F17 Context aware	Services in Algorithmia are not context aware, unless explicitly programmed that way.  We recommend that, if context awareness is needed, this information is supplied in terms of GSIM objects.
F11 Versioning	Versioning comes out of the box. Each version of an algorithm has a unique end point. When a service is referenced without mentioning a version number, the most recent version is used. In this way, the caller has control over when to switch to a new version of a service.
F12/F13 (De)centralized data	Algorithmia does not make any assumptions on this. It offers limited connections to data sources, but currently no connections to data virtualization platforms. Users will need to "put the data somewhere" themselves before running services.  We recommend that users of the algorithmia platform agree on how to put data to the algorithms.
F14 Service integration	Algorithmia offers integration of other services on Algorithmia out of the box, but not integration with services outside of the Algorithmia platform.
F15 Big Data Capable	Algorithmia is not specifically aimed at big data. They offer no out of the box functionality for big data ingestion.  Recommendation; investigate how Algorithmia fits in the UNECE data architecture and which capabilities are needed for it to function on big data.
F19 Resilience	Underneath Algorithmia is a (public) cloud based platform that has resilience build into it (using Docker for containers and Kubernetes for orchestration and deployment).

## Shared Service



Shared service with three included features.

## Shared Service



Shared service with three included features and two locally added features.

# CSPA Catalogue



- Publicly viewable
- Adding and Editing after login
- Grid view list of available services: name, owner, GSBPM, Definition, Specification and Implementation
- Filter options & word cloud
- Detail: Overview, Definition, Specification, Implementation

# CSPA Service Catalogue

<http://catalogue.shared-statistical-services.org/>





# CSPA Catalogue: Challenges

- Populating the catalogue with services
- Implementing existing services
- Maintenance existing services and updating documentation
- Technical maintenance of catalogue
- Adjust to new Features/Requirements concept & adapters
- Adding metrics on usage
- Versioning/implementation versions&experiences
- Granularity & include algorithms?
- Link to Global Artefacts Catalogue

# Benefits from Sharing a Service

- Improved knowledge through feedback from users
- Larger network of contacts
- Additional validation of our products
- Identification of areas for improvement/additional functionalities
- Increased visibility of your products

\* *Wesley Yung and Claude Poirier*

# CSPA Global Artefacts Catalogue

- Not (yet) integrated set of catalogues to facilitate sharing and collaboration
- Align Investment Catalogue, Capability Catalogue, and CSPA Service Catalogue
- Finished investments → Capability Catalogue
- New services developed → Add to CSPA Service Catalogue

# CSPA Wiki

# Links

- CSPA Service Catalogue (<http://catalogue.shared-statistical-services.org/>)
- Common Statistical Production Architecture Wiki
- CSPA Global Artefacts Catalogue
- CSPA Training, Presentations and Videos
  
- ESS Sharing common functionalities in ESS
- ESTAT ESS SERV [ESTAT-ESS-SERV@ec.europa.eu](mailto:ESTAT-ESS-SERV@ec.europa.eu)



## **We Want You For Sharing Tools**

Contact us to join the Sharing Tool Group:  
Rosemary (chair) [rosemary.mcgrath@stats.govt.nz](mailto:rosemary.mcgrath@stats.govt.nz)  
Taeke (UNECE secretariat) [taeke.gjaltema@un.org](mailto:taeke.gjaltema@un.org)



# Sharing Tools Group



- Common Statistical Production Architecture (CSPA)
  - *“CSPA Implementation support and responsive to the needs of the statistical community” (ToR)*
  - Former CSPA implementation Group
- 25-30 members from 16 Statistical Organisations
- Mode of work:
  - Monthly meetings (plenary and sub-groups)
  - Two Sprint sessions
  - Wiki platform, Slack and Google docs
  - ModernStats World Workshop