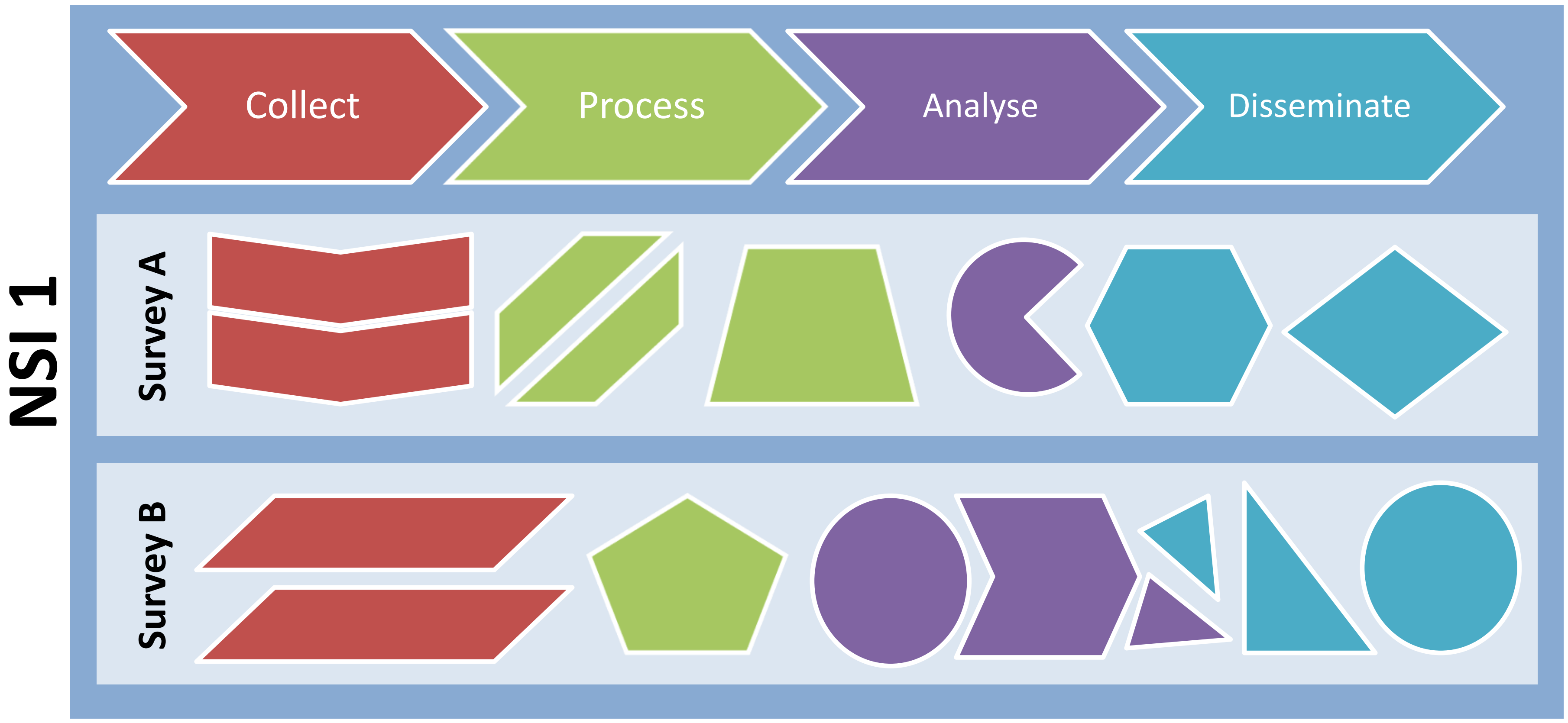
Fostering Interoperability in Official Statistics: Common Statistical Production Architecture

The problem we are trying to solve

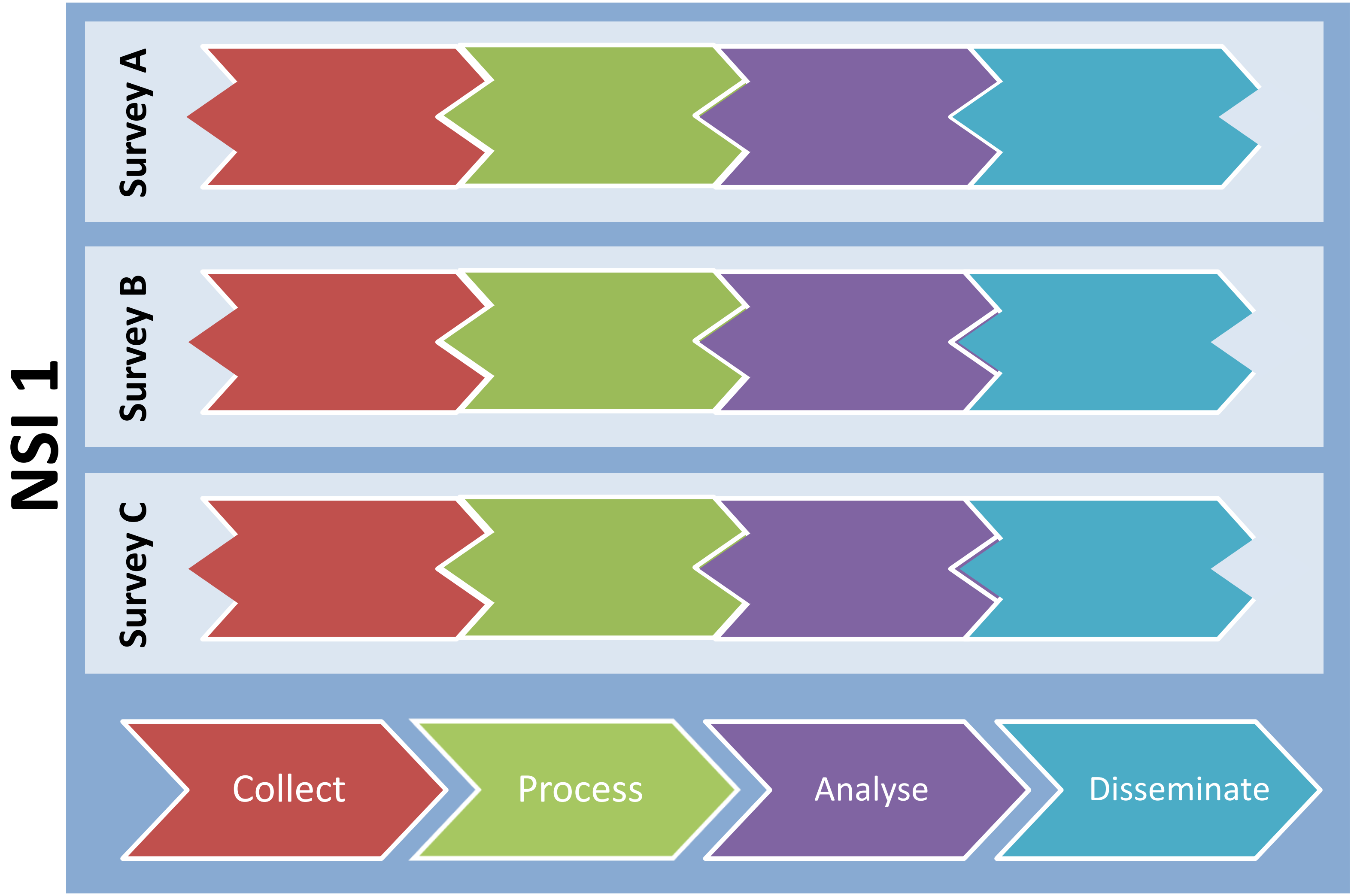


Historically statistical organisations have produced specialised business processes and IT systems

How does Architecture help?

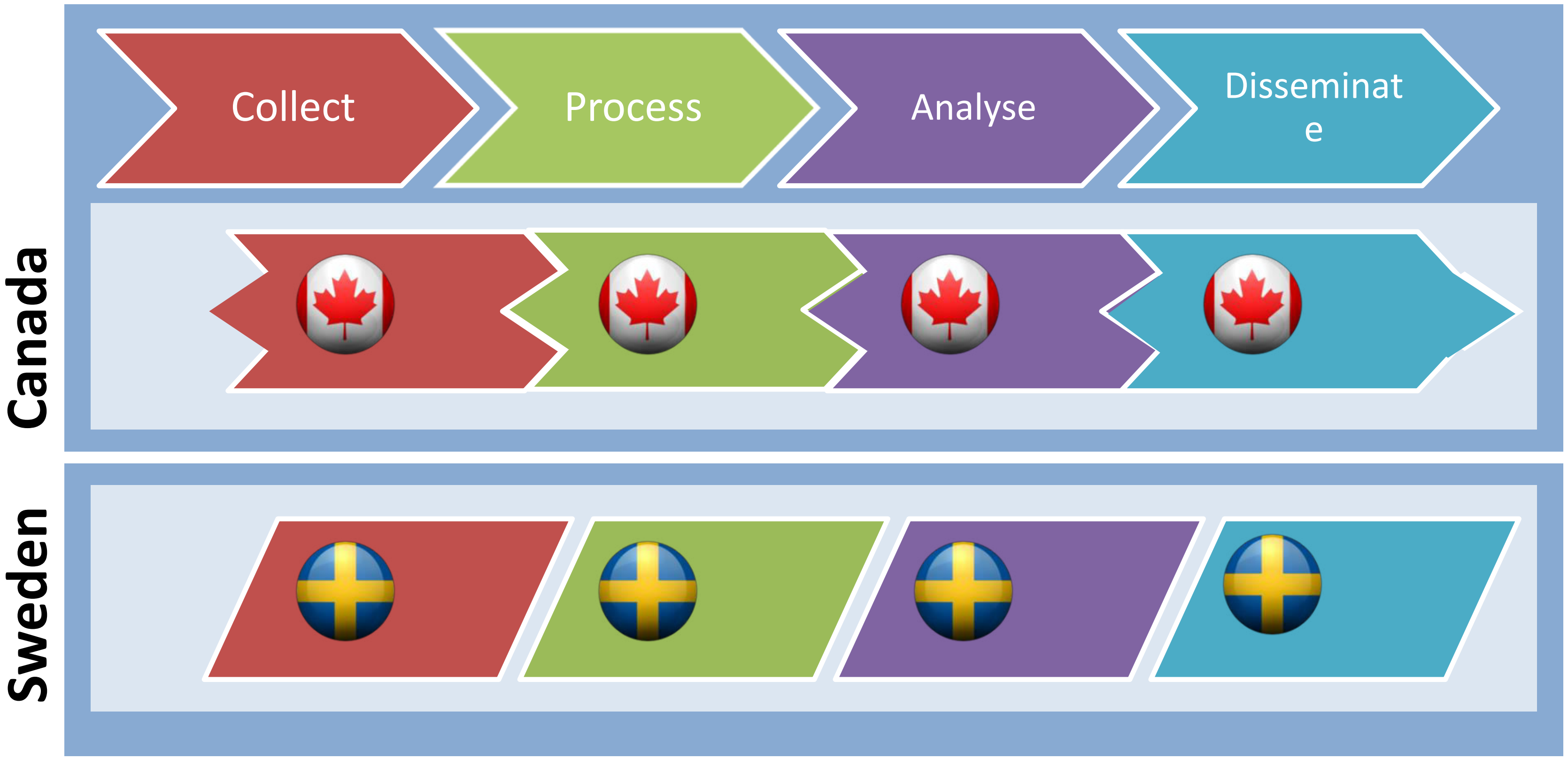
- Many statistical organisations are modernising and transforming using Enterprise Architecture
- Enterprise Architecture shows what the business needs are and where the organisation wants to be, then aligns efforts accordingly
- It can help to remove silos and improve collaboration across an organisation

EA helps you get to this

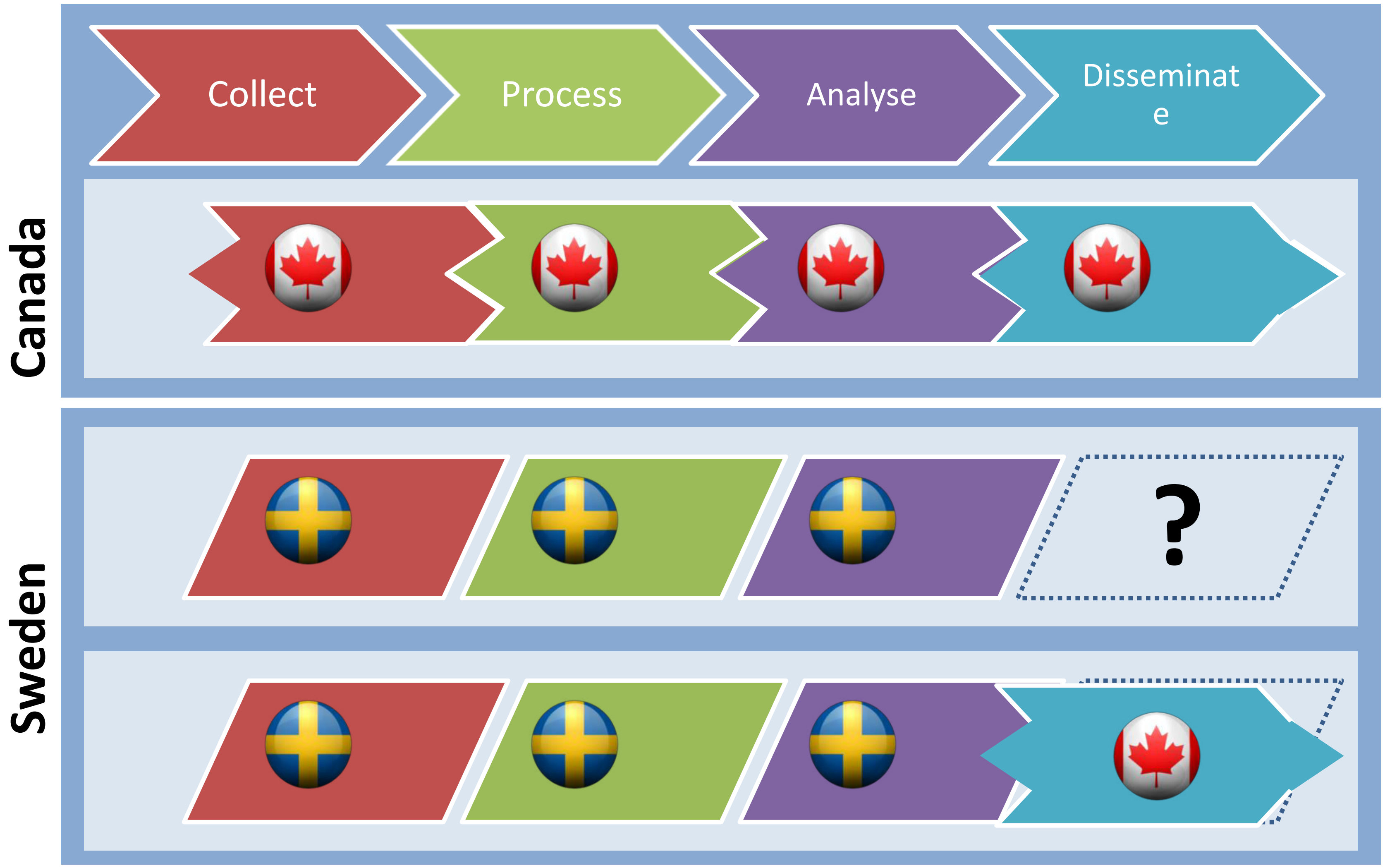


…but if each statistical organisation works by themselves…..

…we get this….

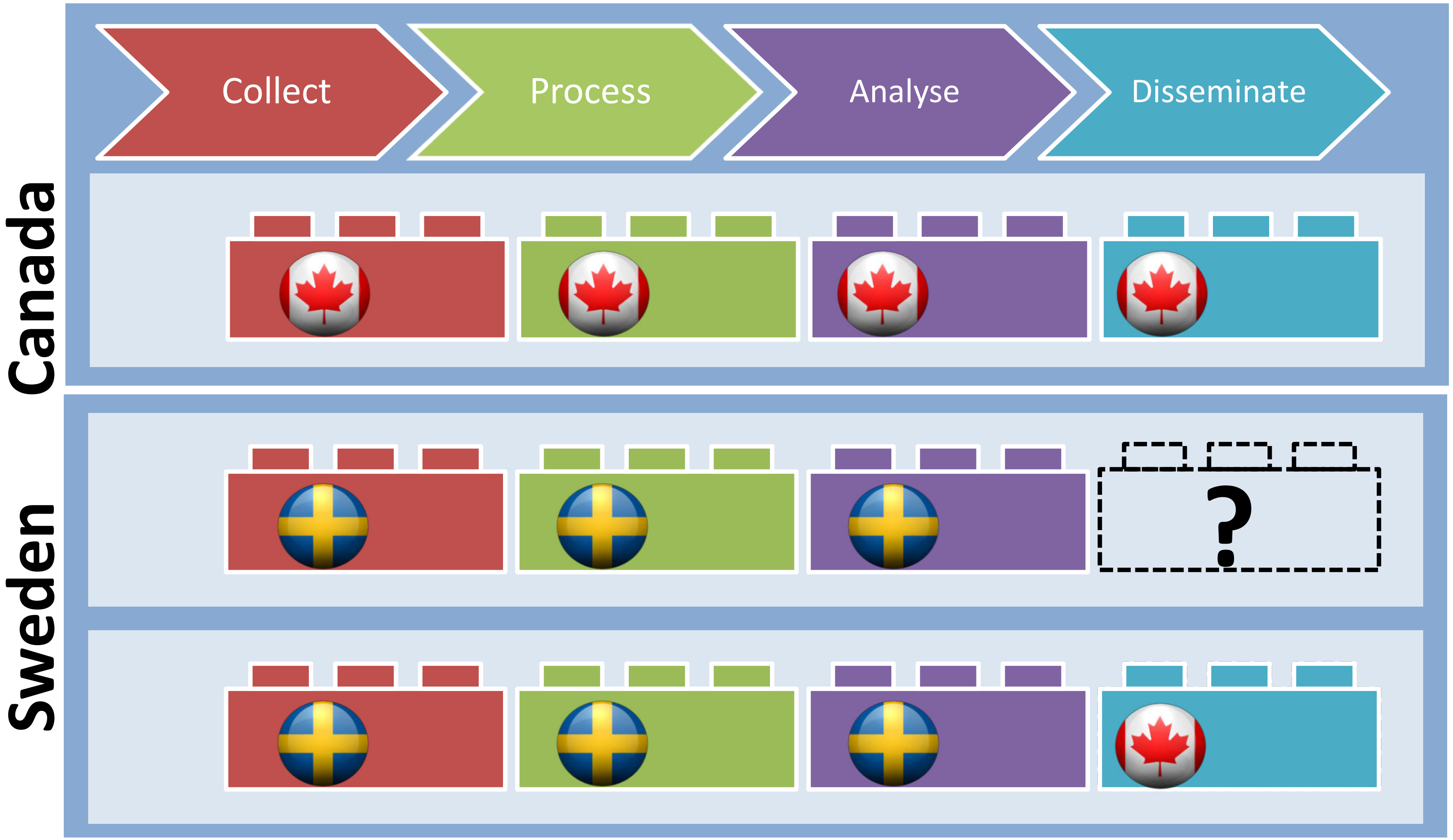


This makes it hard to share and reuse!



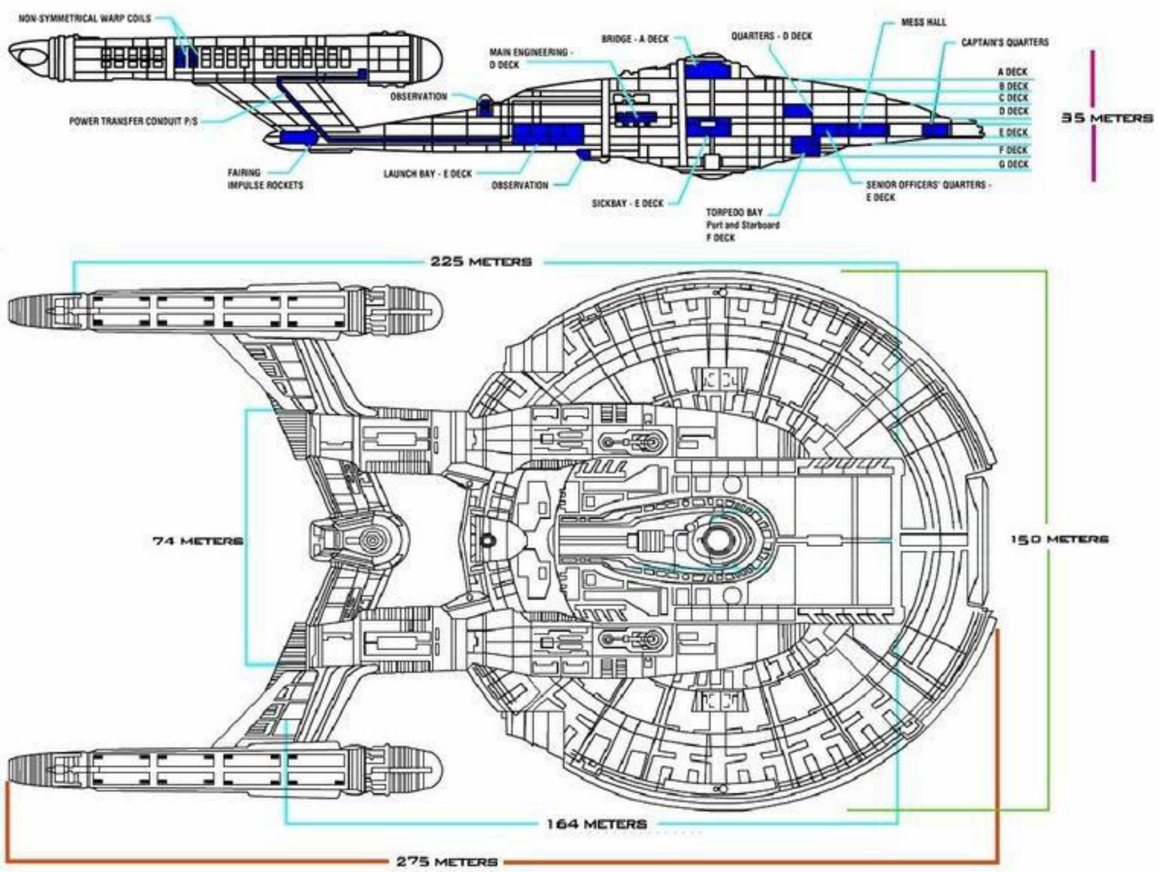
…but if statistical organisations work together?

This makes it easier to share and reuse!



2 Strands to the project

Architecture



Proof of Concept

The Architecture

CSPA Definition

- Common Statistical Production Architecture (CSPA): framework about Statistical Services to create an agreed top level description of the 'system' of producing statistics which is in alignment with the modernization initiative

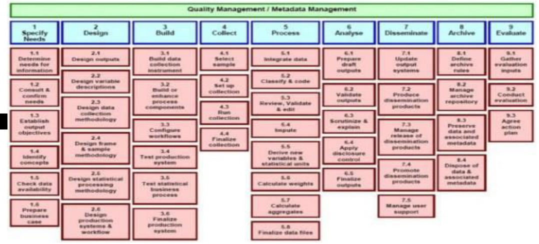
- CSPA provides a template architecture for official statistics, describing:
  - What the official statistical industry wants to achieve
  - How the industry can achieve this, i.e. principles that guide how statistics are produced
  - What the industry will have to do, compliance with the CSPA

Business Architecture

Information Architecture

Application Architecture

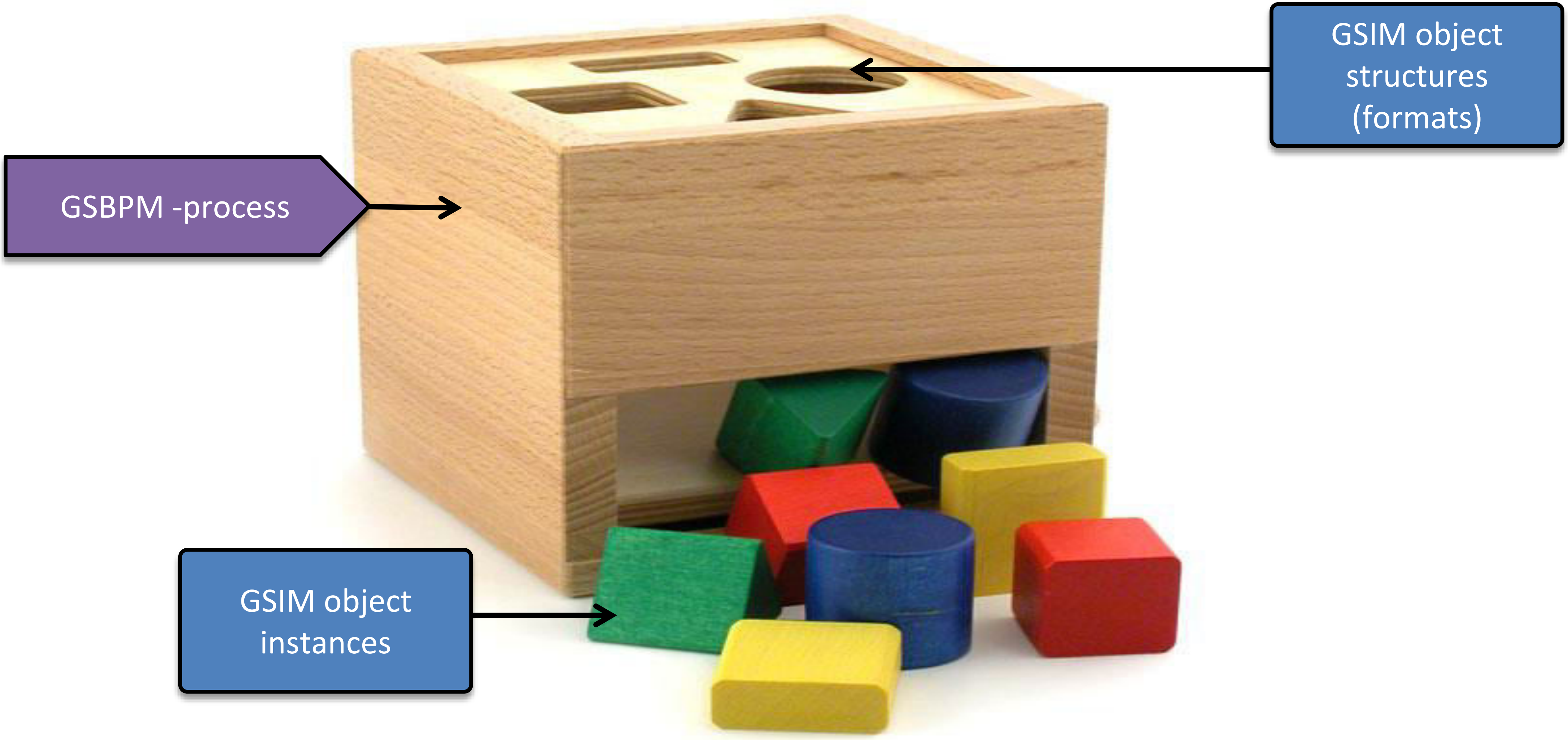
Technology Architecture



<ddi>

sdmx
Statistical Data and Metadata eXchange

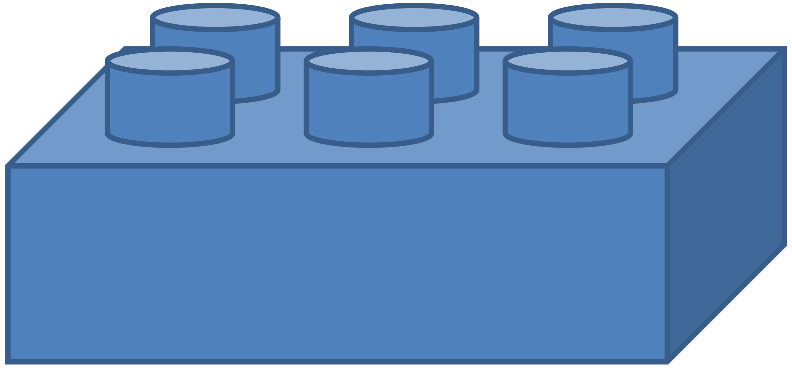
A statistical service



http://www.illeccio.com/page.php?lang=en&product_id=37

The concept of Plug and Play

- Standardised Service:
  - **Standardised** input and output
  - Meet generic nonfunctional **requirements**
  - Can be **easily** used and reused in a number of different processes



Proof of Concept

Choosing the PoC components

Lego pieces could be:

Brand new

OR

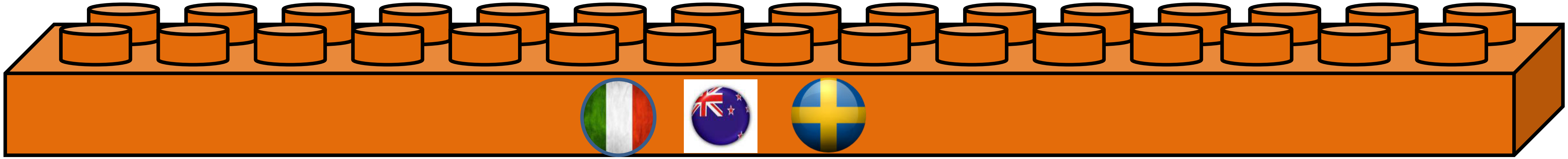
Wrapped legacy/existing

The Proof of Concept

- 5 countries played the role of Builders

Editrules | CANCEIS | Blaise | G Code | SCS

- 3 countries played the role of Assemblers

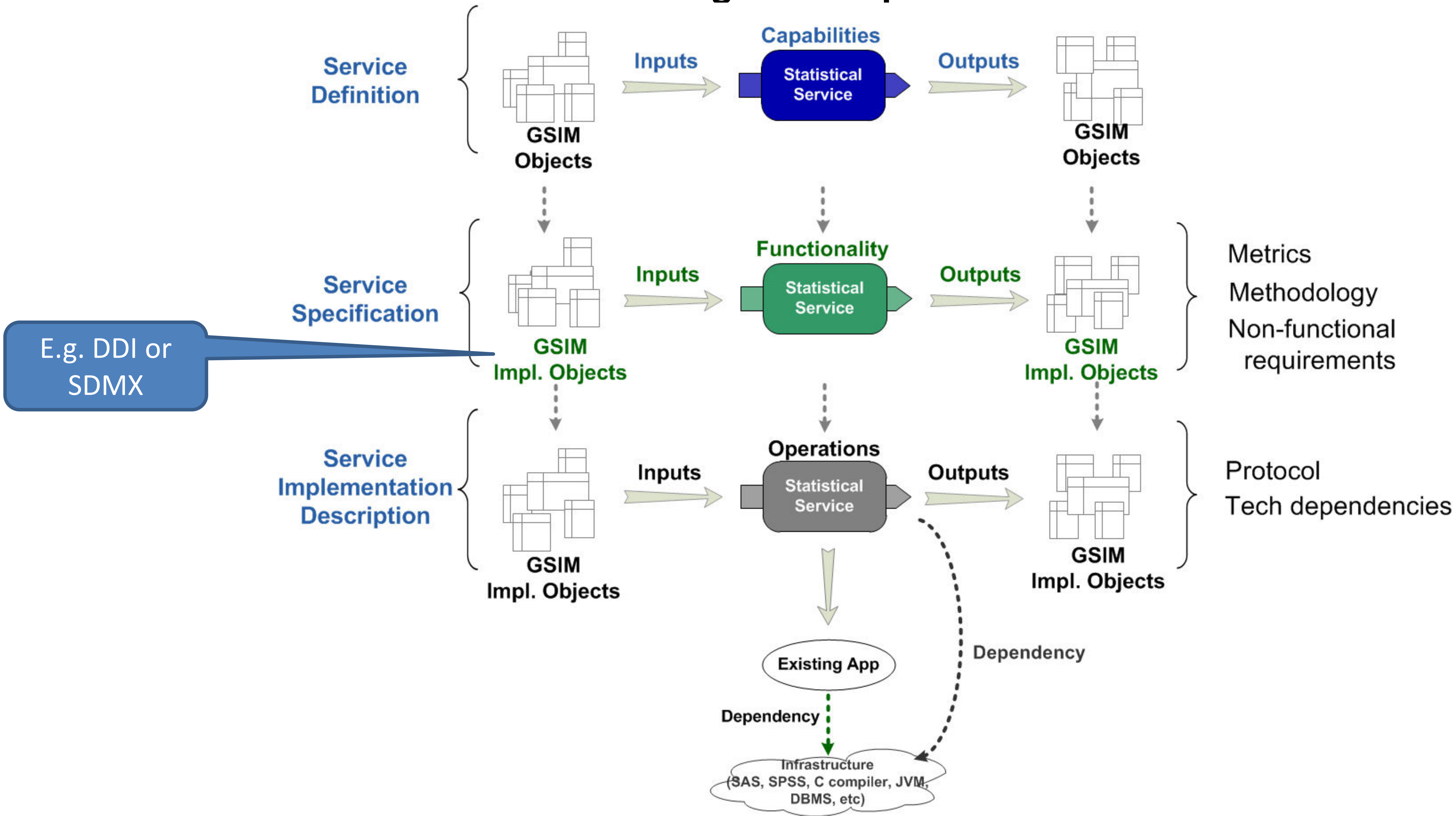


What Did the Services Do?

- DataEdit: Localization of erors
- CANCEIS: Localization of errors, editing, imputation
- Blaise: Administration of questionnaire and collection of data
- G Code: An auto-coding service
- SCS: An auto-coding service

Using DDI in the Proof of Concept

CSPA Service Design and Implementation



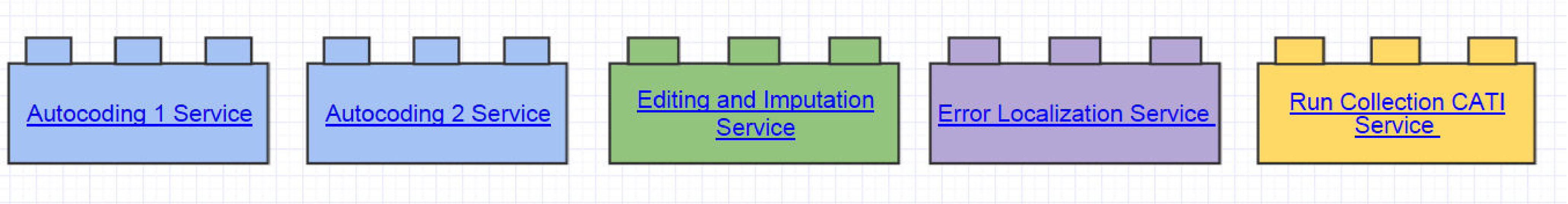
Learning curves

Proof of Concept required knowledge about:

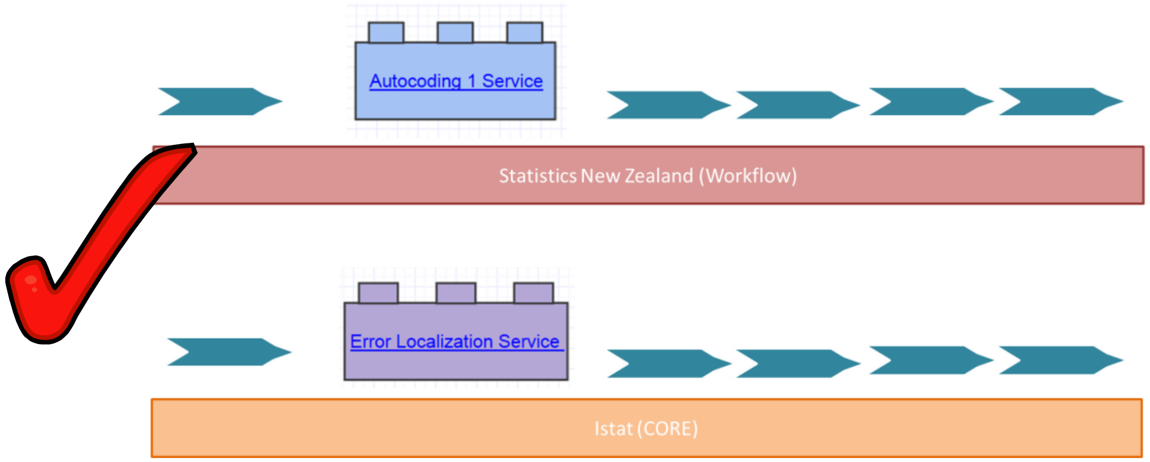
- The tool which was wrapped (CANCEIS, Blaise etc)
- GSIM implementation standards (DDI in this case)

What did we prove?

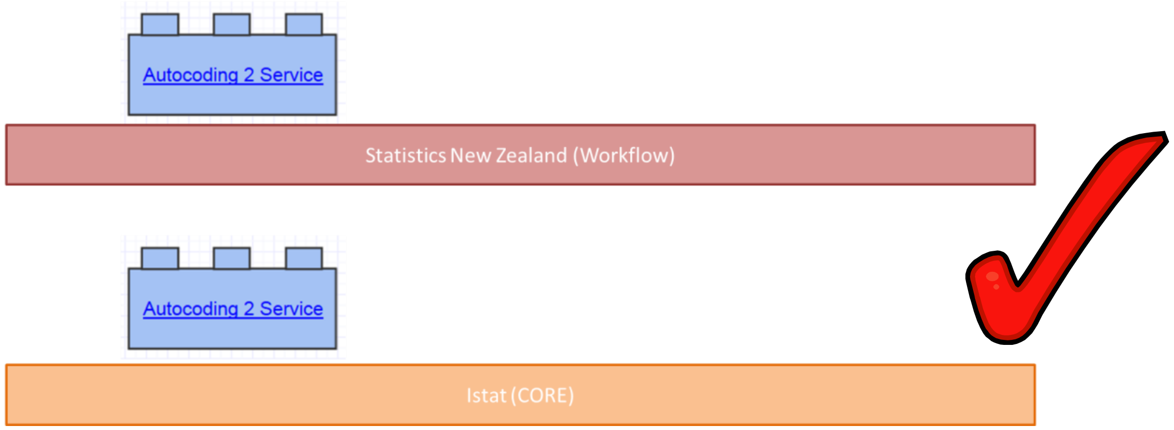
CSPA is practical and can be implemented by various agencies in a consistent way



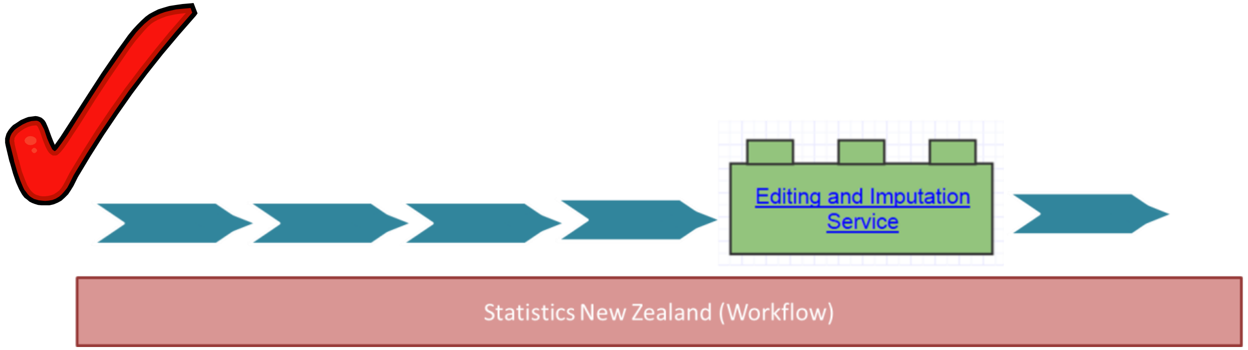
You can fit CSPA Statistical Services into existing processes



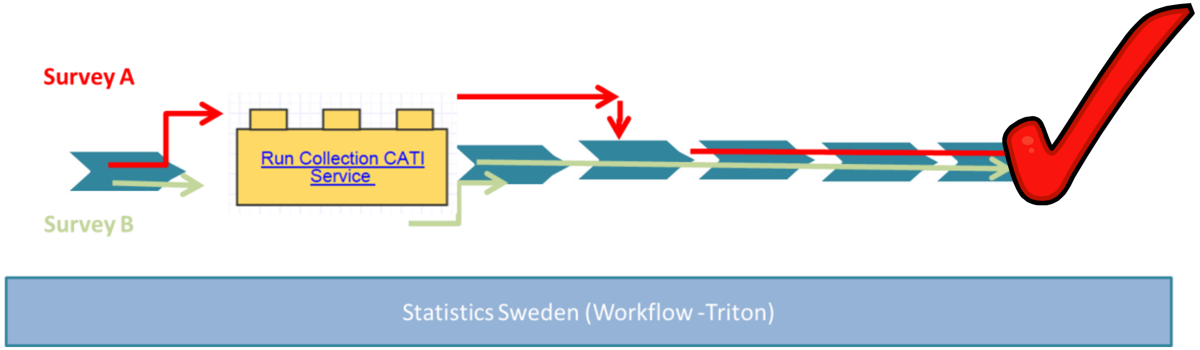
CSPA does not prescribe the technology platform an agency requires



You can swap out CSPA compliant services easily



Reusing the same statistical service by configuration



What was the CSPA POC Experience with DDI?

- Being a lifecycle-oriented project, the CSPA POC agreed to use DDI 3.1, the latest production version of DDI Lifecycle
- The services focused in two areas: questionnaires and (mostly) editing of microdata (re-coding, localization of errors, imputation)
- DDI Lifecycle was the natural choice
  - DDI maps reasonably well to GSIM
  - DDI profiles and "implementers guide" now being produced

DDI Lessons Learned (1)

- For data editing, DDI Lifecycle can be massive overkill
  - Much of the required detail is simply not needed (better in 3.2)
- Data editing is a relatively “metadata-light” application
  - A few data files needed to be described, for input data sets, edited data sets, and reports (tables of which variables were imputed, or where errors might be located)
  - These files were mostly very simple .CSV files
  - We also needed a codelist (codes and categories) for the coding services
- A *really simple* data set description is needed
  - No interest in study-level information: it is not used by these applications
  - This document will be included in DDI 4.* and later

DDI Lessons Learned (2)

- It is important to maintain the continuity of metadata across the lifecycle
- The editing phases of the lifecycle do not use a lot of metadata
  - The tools often consume metadata, but do not produce much! (SAS, etc.)
- Study-level metadata is often fairly static
- Variables, logical records, physical data description, statistics can be "recovered" from post-process set-ups, etc.
- Otherwise, the processing phases of the life-cycle can be a "metadata black hole"!

DDI Lessons Learned (3)

- CSPA as an architecture is services-oriented
  - The definition of services is broad (TOGAF), but web services and RESTful services both fit the definition being used
  - DDI is not service-oriented: there are no standard service interfaces
- Most “files” were passed into the CSPA POC services as location references
  - DDI was passed in wholesale in XML form
  - This would not be necessary is we had a standard RESTful syntax, etc.
  - Metadata could be obtained as needed by the services at run-time from minimal input parameters

DDI Lessons Learned (4)

- The CSPA architecture is designed to support more than just data-production processes
  - Also “support” functions such as classification management
- In GSIM, the Study Unit maps neatly to a cycle of data production
  - There is no good corresponding container for support functions: Study Unit is about data production
  - Resource Packages represent reusable resources, and map against other things in GSIM
- For the CSPA POC, this was not an issue: all services were data-oriented

An Interesting Decision: Rules Language

- For the CSPA POC, many GSIM inputs were “Rules”
  - For imputation
  - For editing
  - For validation
- There was no good “rules language” for expressing these in a standard way
- Decision was made, for future work, to use the platform-neutral “Expression Language” now being developed
  - For use with SDMX and DDI, or as “stand-alone”
  - Second face-to-face meeting will be in Basel, end of January 2014

Summary

- DDI was able to support the CSPA POC use cases
- Too complex, and too steep a learning curve
- Standard DDI services interfaces should be developed
- Need to think about the overall data production lifecycle and how to persist the metadata
- Need to consider the GSIM objects not only for cyclical data production, but also for “support” functions such as metadata management