Future of Federations

(Technical)

Ian A. Young

UK federation, Shibboleth Consortium

ian@iay.org.uk

Prediction is very difficult,
especially about the future.

– Niels Bohr

(attributed)

The software doesn't know about federations

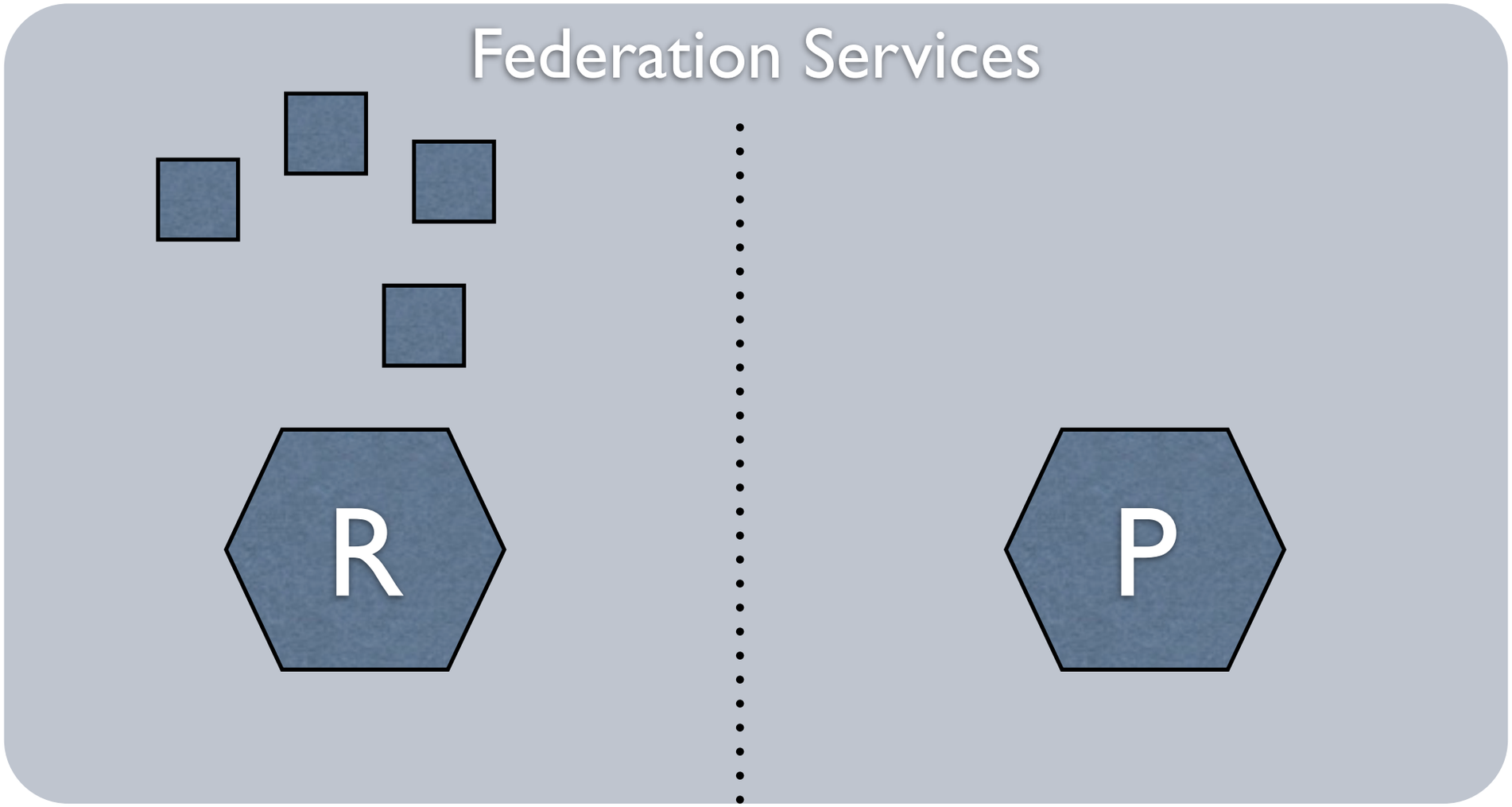
– Scott Cantor

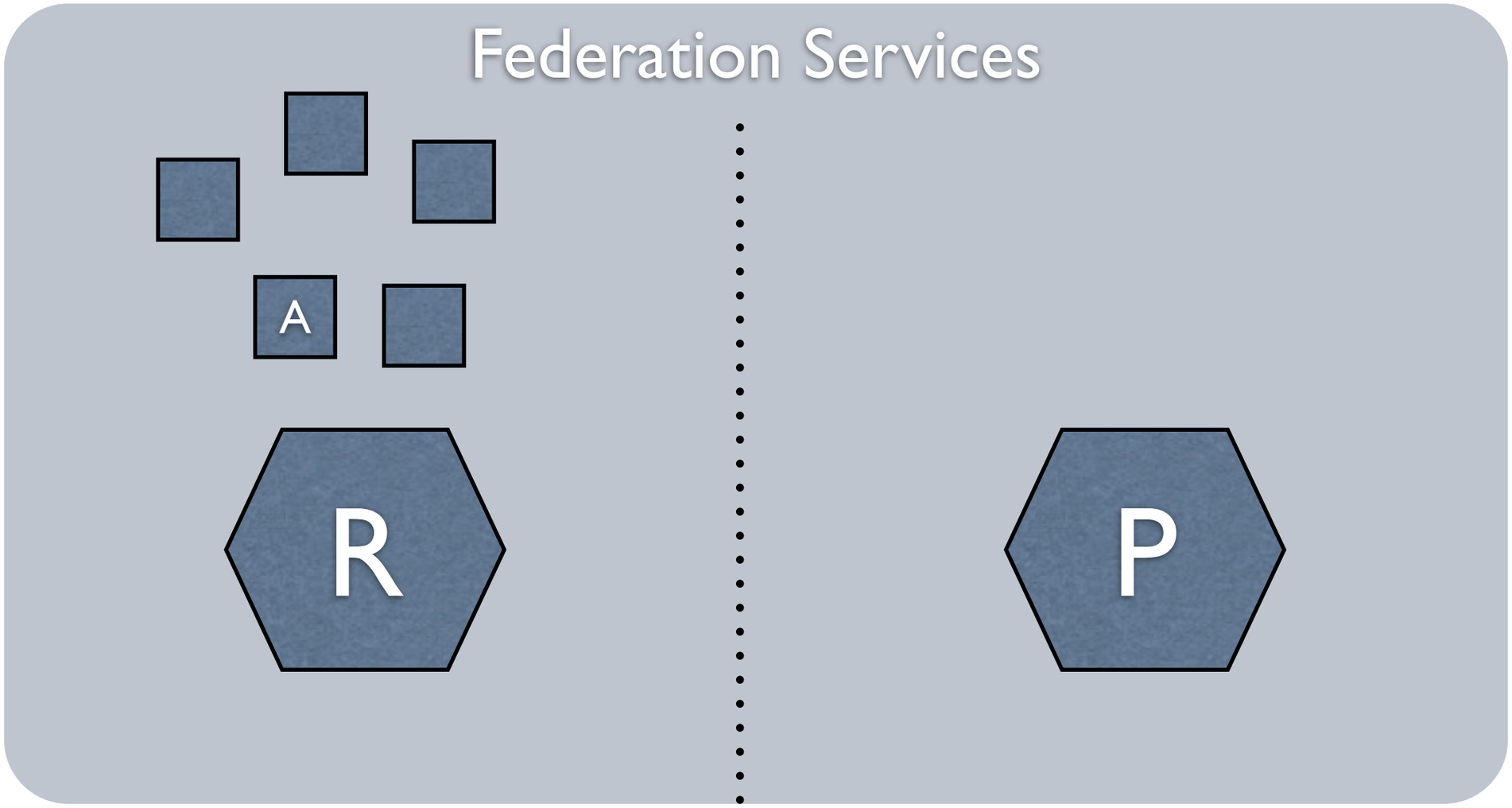
(repeatedly)

Technical Services

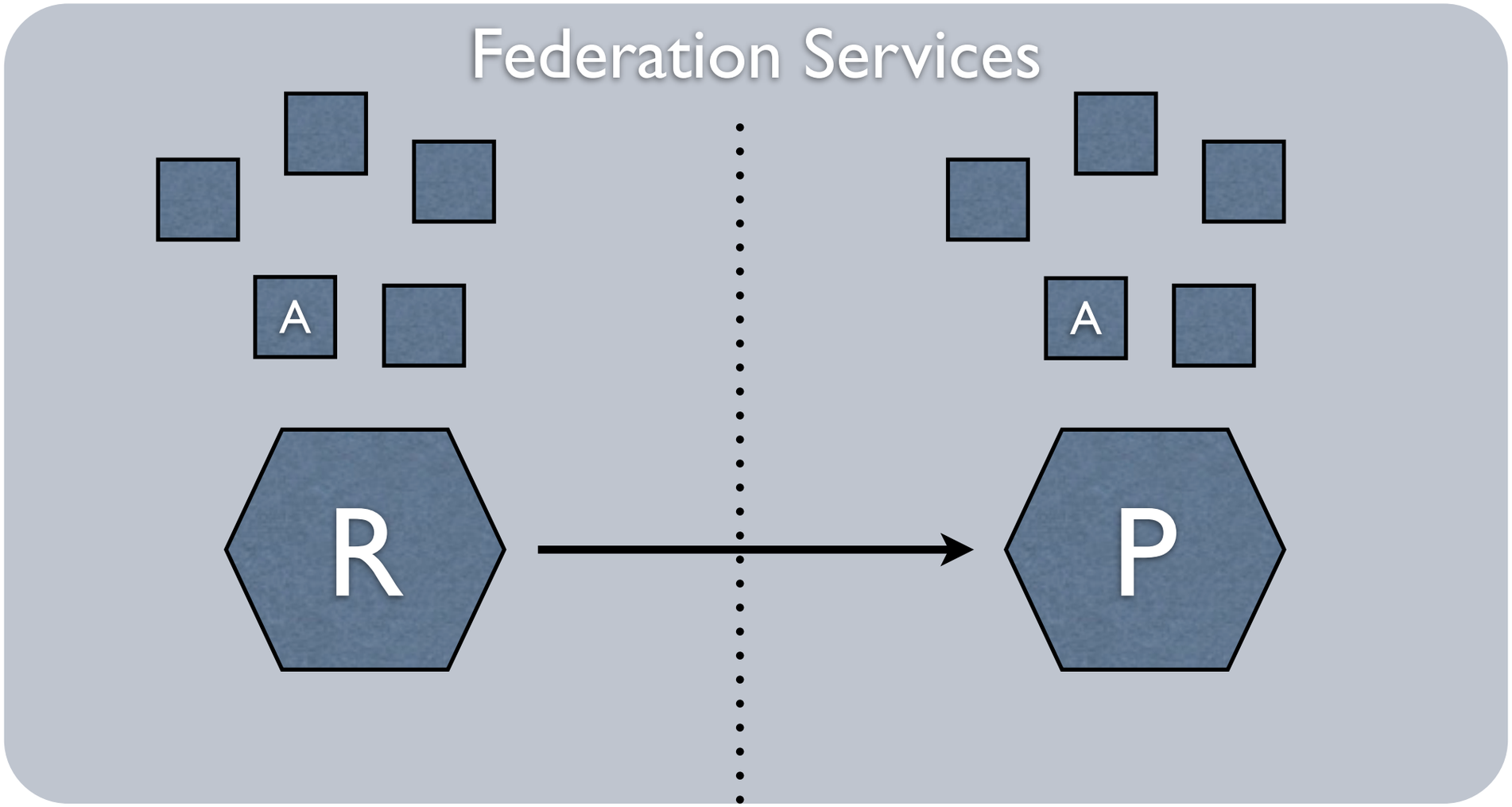
- Federations are *not* primarily technical constructs
- *but* current federations provide a bundle of technical *services* to their members
  - Registration
  - Aggregation
  - Publication

} Inter-federation by metadata exchange

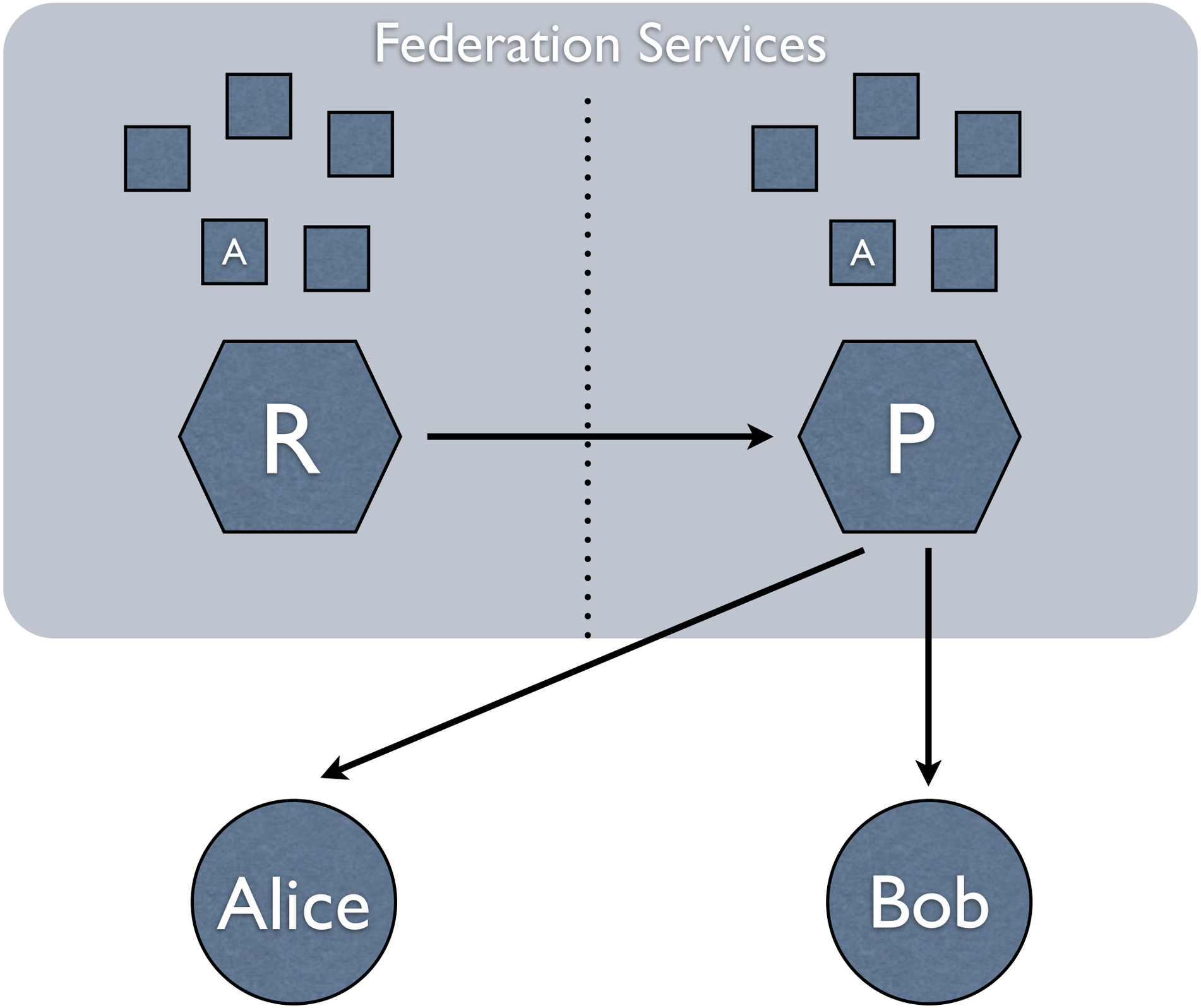

Federation Services
R
P

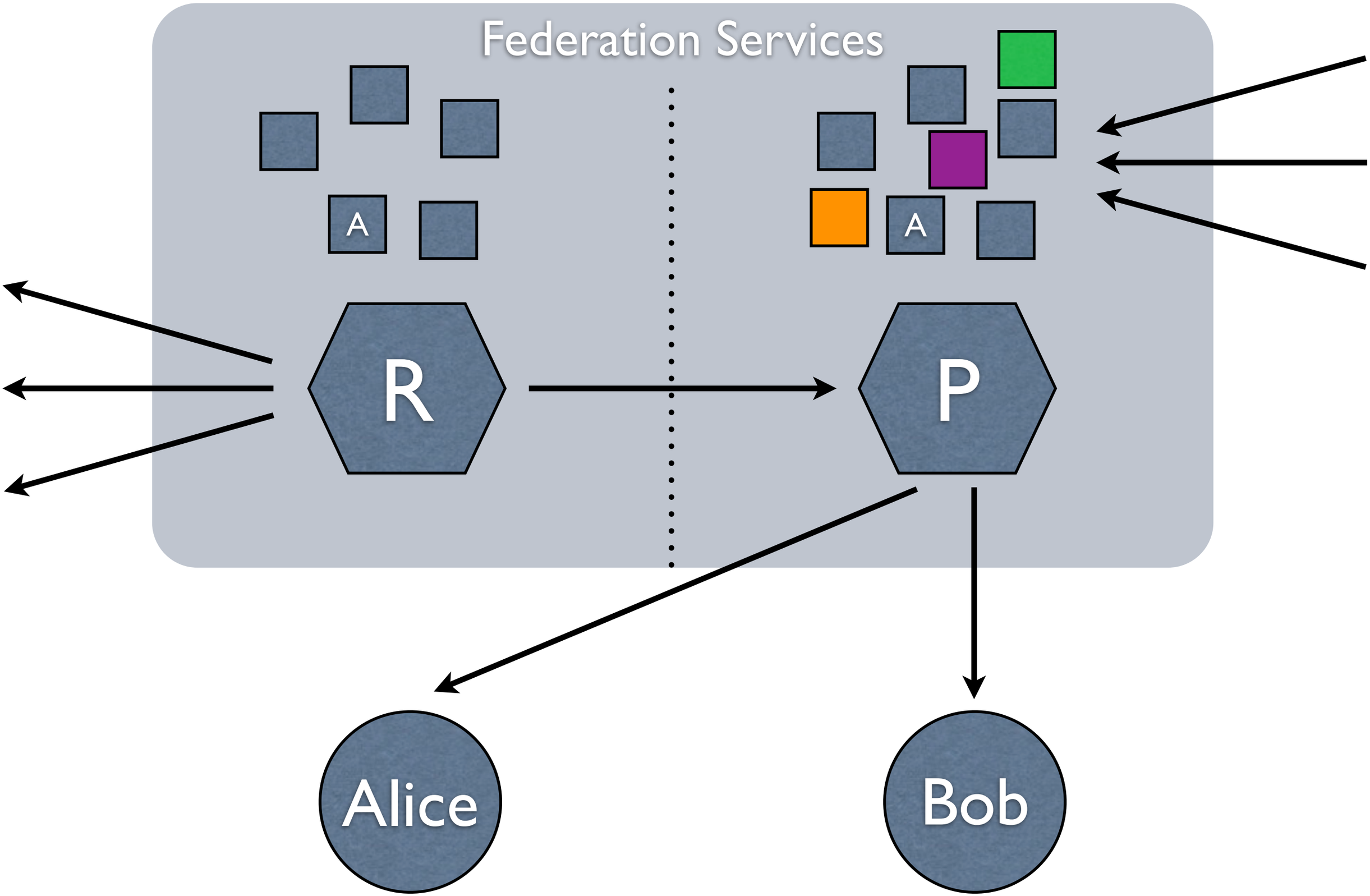

Federation Services
A
R
P


Alice


Federation Services
A
A
R
P


Alice


Federation Services
A
A
R
P
Alice
Bob


Federation Services
A
A
R
P
Alice
Bob

Registration

How we register

- Today, with few exceptions, national federations:
  - perform their own registration
    - use and maintain custom, integrated tools
- This can be seen as redundant and inflexible
- PEER and REEP might change things

PEER

- PEER is a software component designed to do registration and only registration
- Common, modular, implementation means more likely to keep up with latest standards
- Application: registration front end for a federation's registrar service
  - could be internal or public facing
- See https://github.com/Yaco-Sistemas/peer/

REEP

- REEP will be a public deployment of PEER with a very simple publication back end
- A new thing: an independent registrar
- Register or update in one place, not many
- Still likely to require multiple membership
- See https://refeds.terena.org/index.php/PEER_FAQ

What we register

- Trend towards much richer metadata:
  - MDUI (discovery UI)
  - MDRPI (registration information)
  - Attribute requirements
  - Algorithm agility metadata
  - Entity attributes: labels, tags, categories

Algorithm Agility

- Academic cryptographers are clever chaps
- Our cryptography is not as strong as we thought; we need to move forward
- Add metadata to say what entities support
- ```
  <alg:DigestMethod
    Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
  ```
- Auto-generation of this metadata is critical
- We have time, but no-one knows how much.
  ```

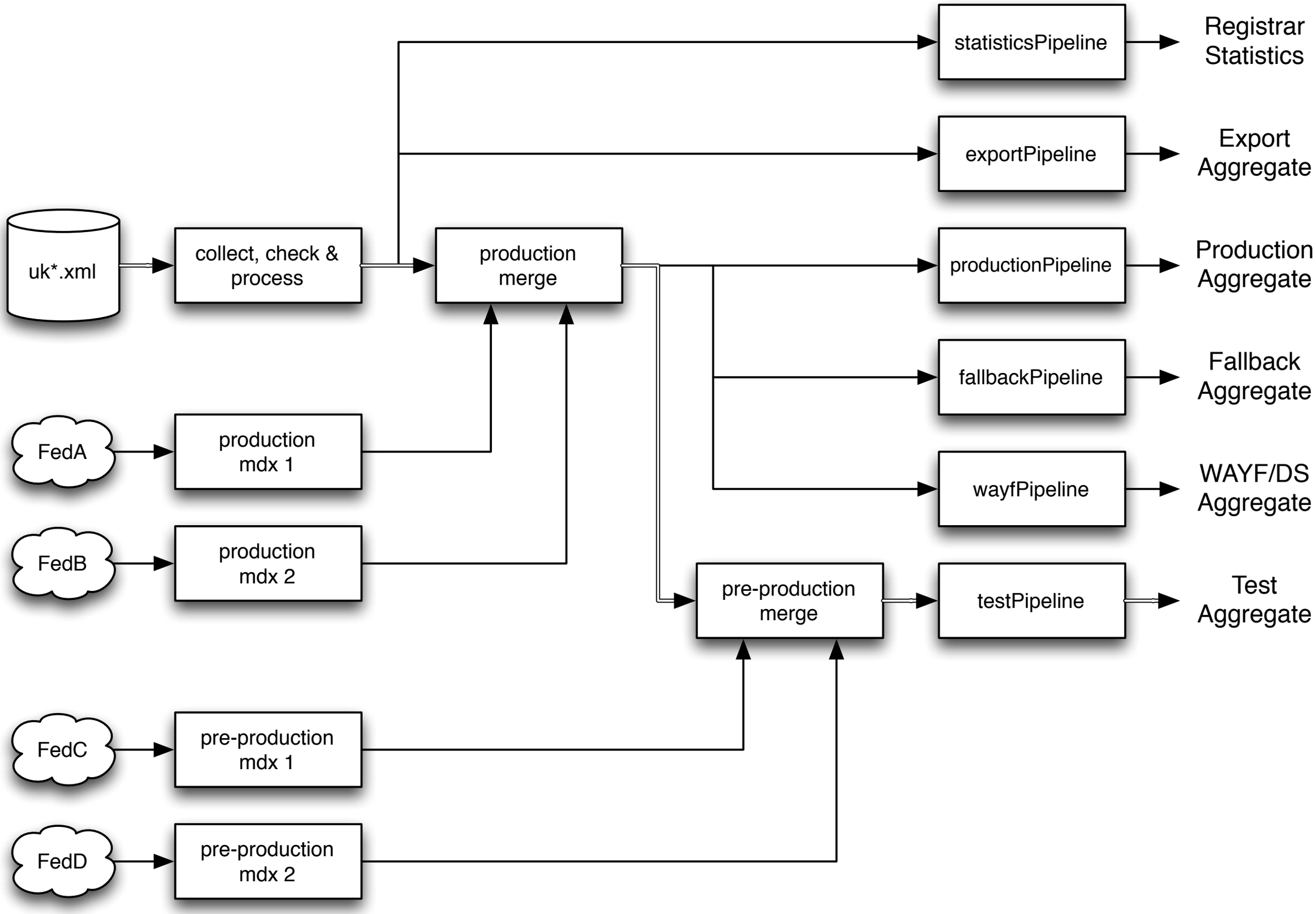
Categories

- Specification at http://macedir.org/
- Built on top of entity attributes specification
- Defines entity category attribute and entity category support attribute
- Anyone can define categories (URI named)
- Initial deployment used for R&S in InCommon, see https://spaces.internet2.edu/x/-oKVAQ

Aggregation

Aggregation Engines

- Is aggregation just smashing entities together?
- Aggregation engines are a more flexible idea:
  - Process *items* through *pipelines* of *stages*
  - Stages can filter, transform, check for errors or of course aggregate
- Implementations: Shibboleth MDA, pyFF


uk*.xml
collect, check & process
production merge
statisticsPipeline
Registrar Statistics
exportPipeline
Export Aggregate
productionPipeline
Production Aggregate
fallbackPipeline
Fallback Aggregate
wayfPipeline
WAYF/DS Aggregate
pre-production merge
testPipeline
Test Aggregate
FedA
production mdx 1
FedB
production mdx 2
FedC
pre-production mdx 1
FedD
pre-production mdx 2

Further Reading

- Some Notes on Metadata Interchange:
  - http://www.iay.org.uk/blog/2008/10/metadata_interc.html
- Concepts and Methods:
  - http://www.iay.org.uk/blog/2009/05/concepts_and_me.html
- UK Federation Metadata Aggregation:
  - http://iay.org.uk/blog/2012/08/uk-federation-metadata-aggregation

Publication

Metadata Query

- Publishing bulk aggregates has long term problems: size, brittleness
- Solution: lightweight per-entity query model (MDQuery, sometimes called MDX)
- Currently an expired I-D (draft-lajoie-md-query-01); this is being revived
- Client software support thin at present
- InCommon and UK federation considering trials

Questions?