eosc | FAIR-EASE

The Mythical Data Lake
From Data Lake to Data Spaces

Sept 27, 2023 @OSFAIR

Katrina Exter (VLIZ), Marc Portier (VLIZ)

Funded by
the European Union

eosc | FAIR-EASE

The need for a Data Lake in FAIR-EASE

The services FAIR-EASE will build

- DDAS: Data Discovery and Access Service. Users and clients will be able to
  - Find and access their own data and associated/related data
  - Find and access subsets within datasets
- VRE: Virtual Research Environment. Users will be able to
  - Call up software and services to work on data
  - Save/export the outputs of those services
  - Build up workflows/notebooks

So the FE environment needs to

- know where the data are (DDAS)
- know how to interrogate the data so they can be searched, accessed, subsetted, quantified, … (DDAS, VRE)
- know how to move those data into the services (VRE)
- know how to incorporate the data output from the services back into the DDAS and VRE

eosc | FAIR-EASE

The need for a Data Lake in FAIR-EASE

The services FAIR-EASE will build

- DDAS: Data Discovery and Access Service. Users and clients will be able to
  - Find and access their own data and associated/related data
  - Find and access subsets within datasets
- VRE: Virtual Research Environment. Users will be able to
  - Call up software and services to work on data
  - Save/export the outputs of those services
  - Build up workflows/notebooks

So the FE environment needs to

- know where the data are (DDAS)
- know how to interrogate the data so they can be searched, accessed, subsetted, quantified, … (DDAS, VRE)
- know how to move those data into the services (VRE)
- know how to incorporate the data output from the services back into the DDAS and VRE

The need for a Data Lake in FAIR-EASE

The services FAIR-EASE will build

- DDAS: Data Discovery and Access Service. Users and clients will be able to
  - Find and access their own data and associated/related data
  - Find and access s
- VRE: Virtual Research
  - Call up software a
  - Save/export the d
  - Build up workflows/notebooks

DATA LAKE

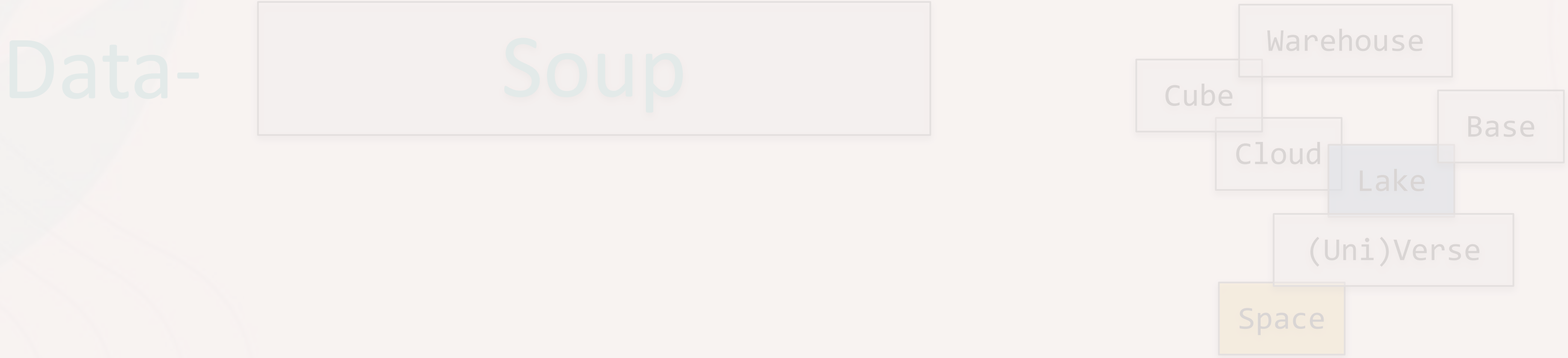
So the FE environment needs to

- know where the data are (DDAS)
- know how to interrogate the data so they can be searched, accessed, subsetted, quantified, … (DDAS, VRE)
- know how to move data into the services (VRE)
- know how to incorporate the data output from the services back into the DDAS and VRE

What is the FAIR-EASE “Data Lake”?

An undefined “somewhere” that “somehow” would allow the DDAS and VRE to find, search, access, subset, and otherwise do whatever we want with FAIR-EASE data

An amorphous blob with datasets swimming around in it?
A single, huge dataset that all individual datasets have “bled into” ?
A place where datasets bob about waiting to be fished out?



Data- Soup

Warehouse

Cube

Base

Cloud

Lake

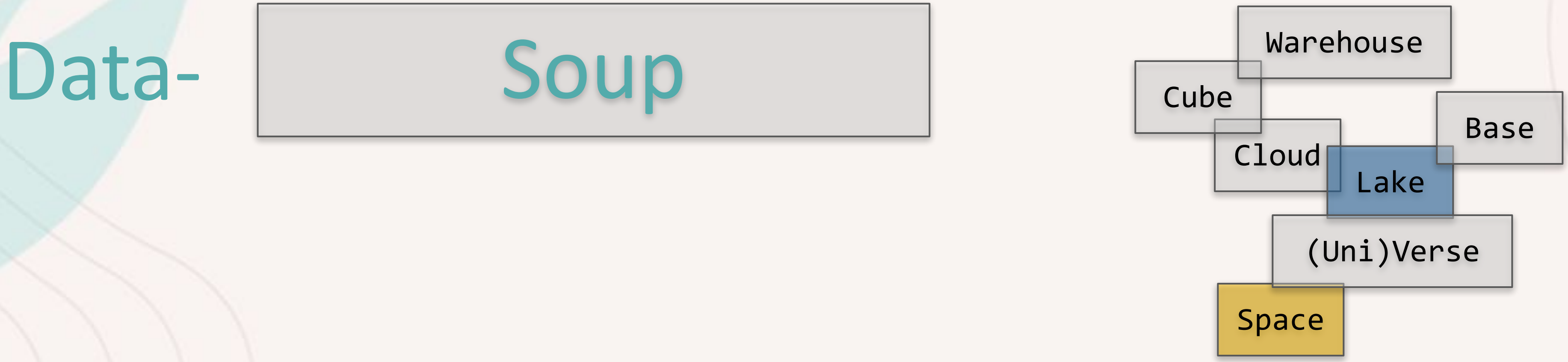
(Uni)Verse

Space

What is the FAIR-EASE "Data Lake"?

An undefined "somewhere" that "somehow" would allow the DDAS and VRE to find, search, access, subset, and otherwise do whatever we want with FAIR-EASE data

An amorphous blob with datasets swimming around in it?
A single, huge dataset that all individual datasets have "bled into" ?
A place where datasets bob about waiting to be fished out?



Data- Soup

Warehouse

Cube

Base

Cloud

Lake

(Uni)Verse

Space

What is the FAIR-EASE “Data Lake”?

Requirements

1. To allow all FAIR-EASE data to be findable and accessible
2. To allow sharing of data with the community
3. To allow for search and access to subsets and recombinations of data
4. To provide download management and manage user authentication
5. To be flexible to evolving data sets and requirements

Restrictions

1. The diversity of data (types, sizes, accessibility, formats) is enormous and evolving
2. We need to be sustainable: whatever is developed, must continue to exist after FAIR-EASE ends
3. We want a solution that can be adopted by others for their own projects
4. We do not and cannot copy data from source to our data lake

What is the FAIR-EASE “Data Lake”?

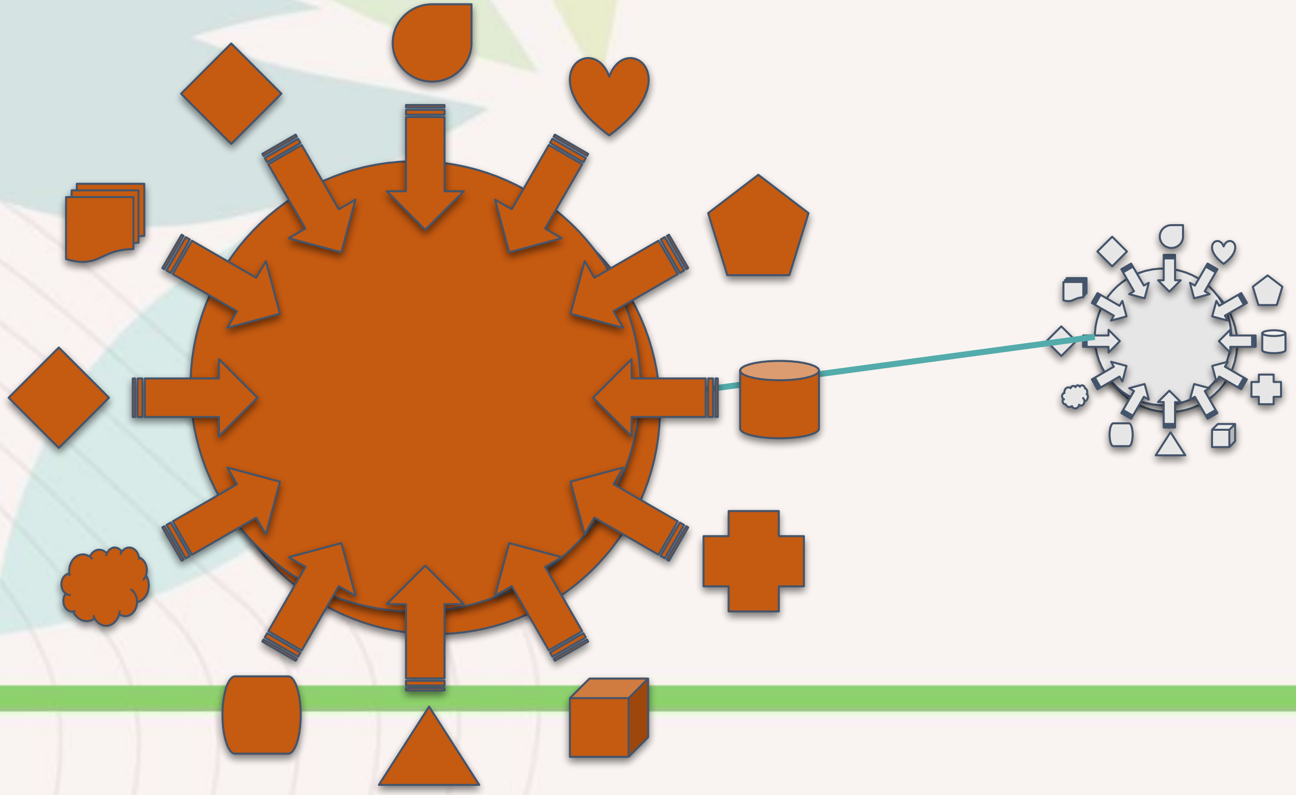
Requirements

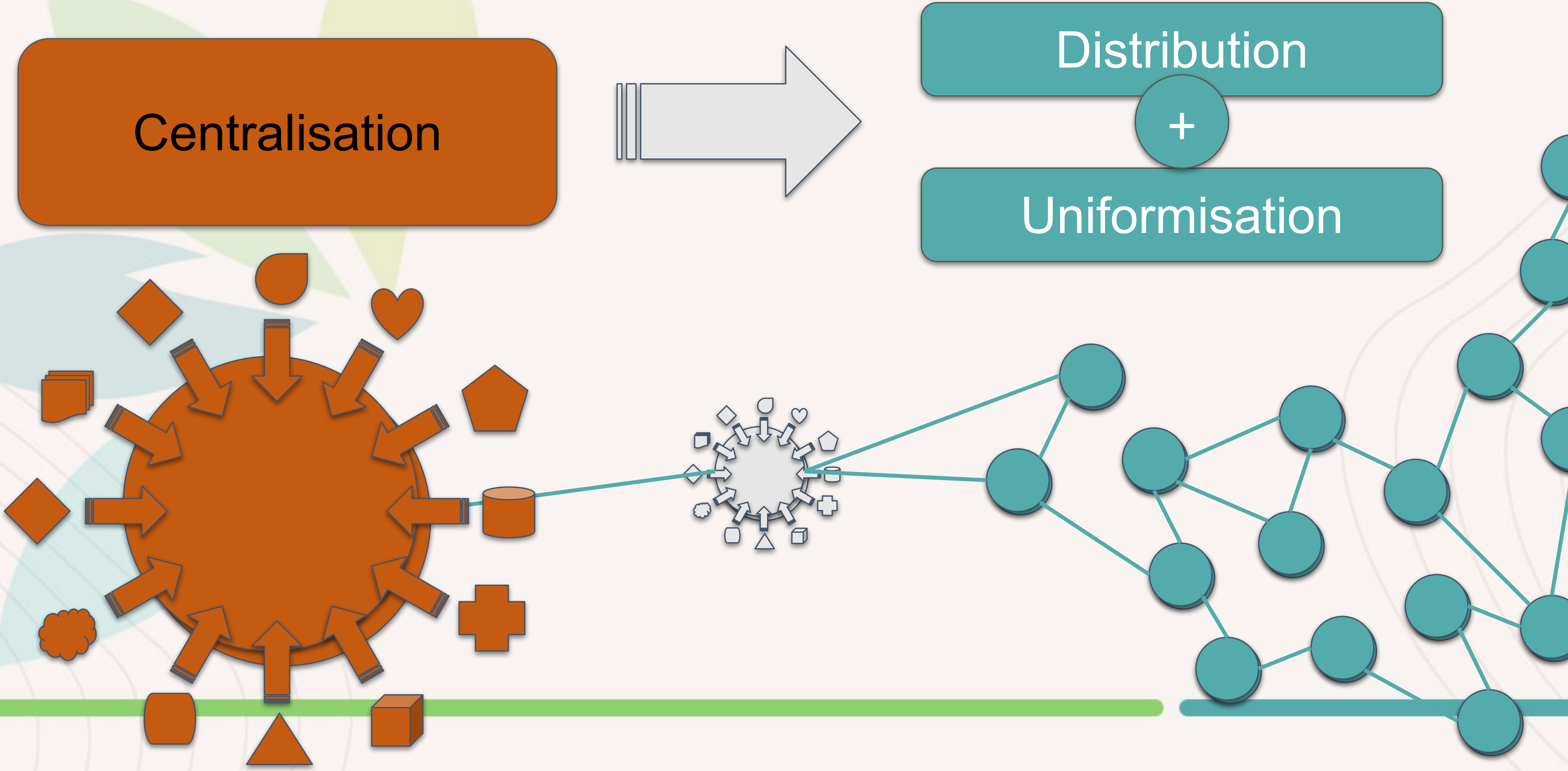
1. To allow all FAIR-EASE data to be findable and accessible
2. To allow sharing of data with the community
3. To allow for search and access to subsets and recombinations of data
4. To provide download management and manage user authentication
5. To be flexible to evolving data sets and requirements

Restrictions

1. The diversity of data (types, sizes, accessibility, formats) is enormous and evolving
2. We need to be sustainable: whatever is developed, must continue to exist after FAIR-EASE ends
3. We want a solution that can be adopted by others for their own projects
4. We do not and cannot copy data from source to our data lake

Centralisation





Centralisation

Distribution

+

Uniformisation

What is the FAIR-EASE “Data Lake”?

The data can remain *where* they are

The data can remain *as* they are

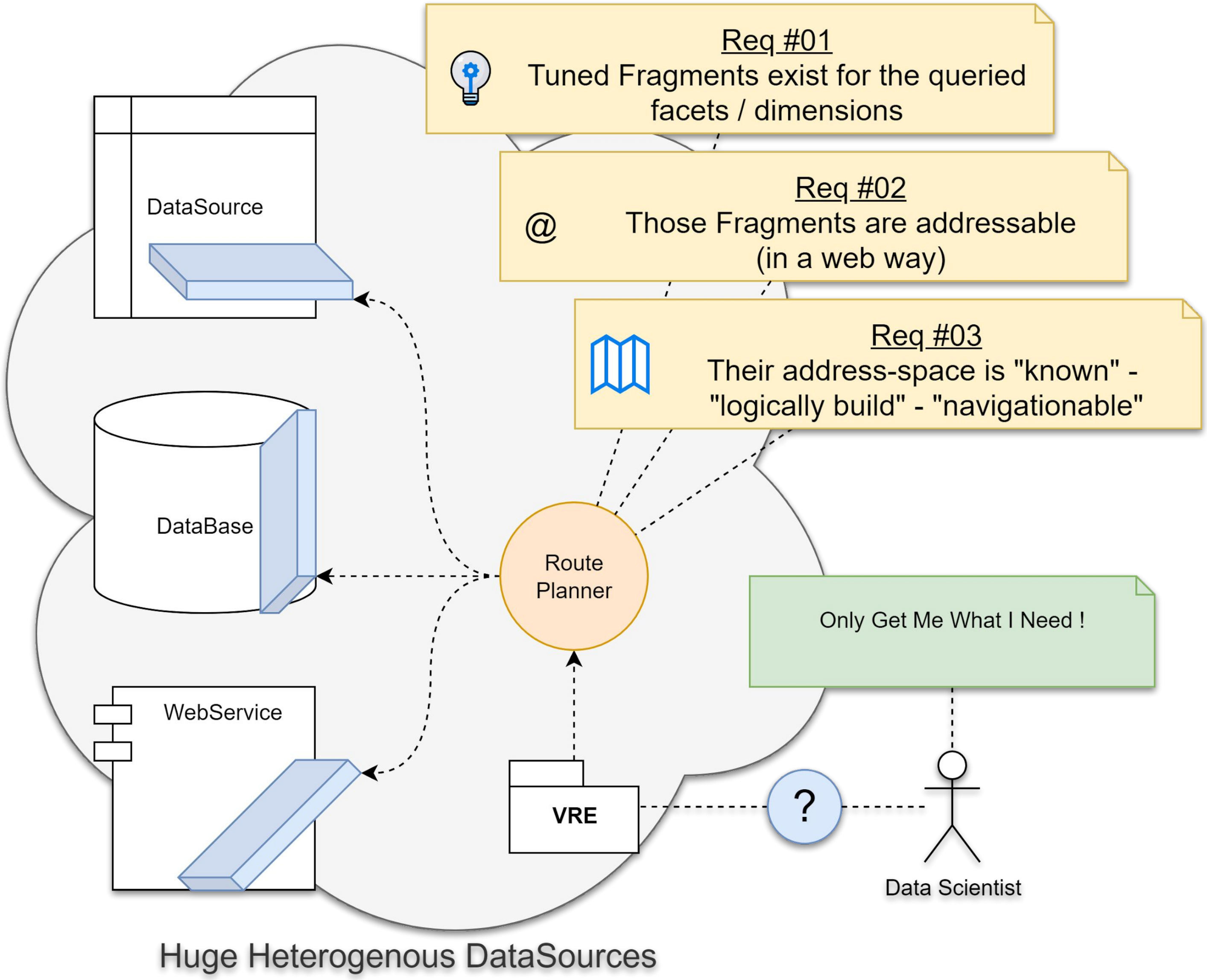
But, a layer of uniformisation sits on top of the data that allows the same findabiliity, accessibility, interoperability, and re-usability for all the data

The Data Lake is not a single place where data are stored

The Data Lake is instead the specifications necessary to add this uniformisation layer

What do you really want?

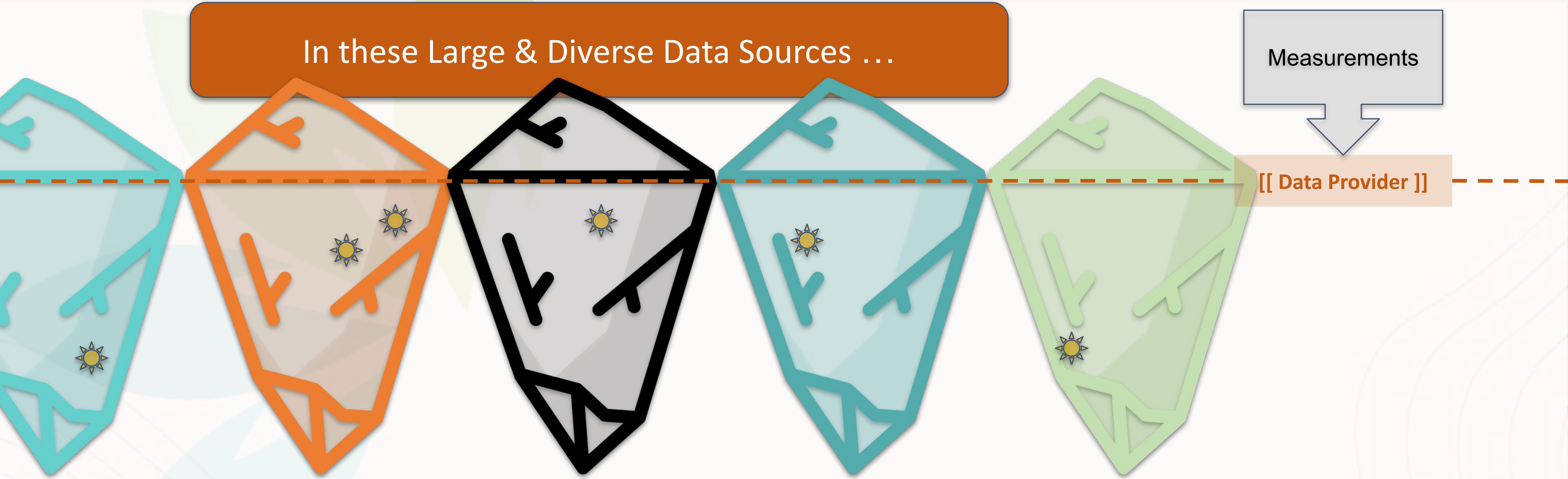
Finding
the shortest path
to only the relevant
information parts.


Req #01
Tuned Fragments exist for the queried facets / dimensions
Req #02
@
Those Fragments are addressable (in a web way)
Req #03
Their address-space is "known" - "logically build" - "navigationable"
DataSource
DataBase
WebService
Route Planner
Only Get Me What I Need !
VRE
?
Data Scientist
Huge Heterogenous DataSources

eosc | FAIR-EASE

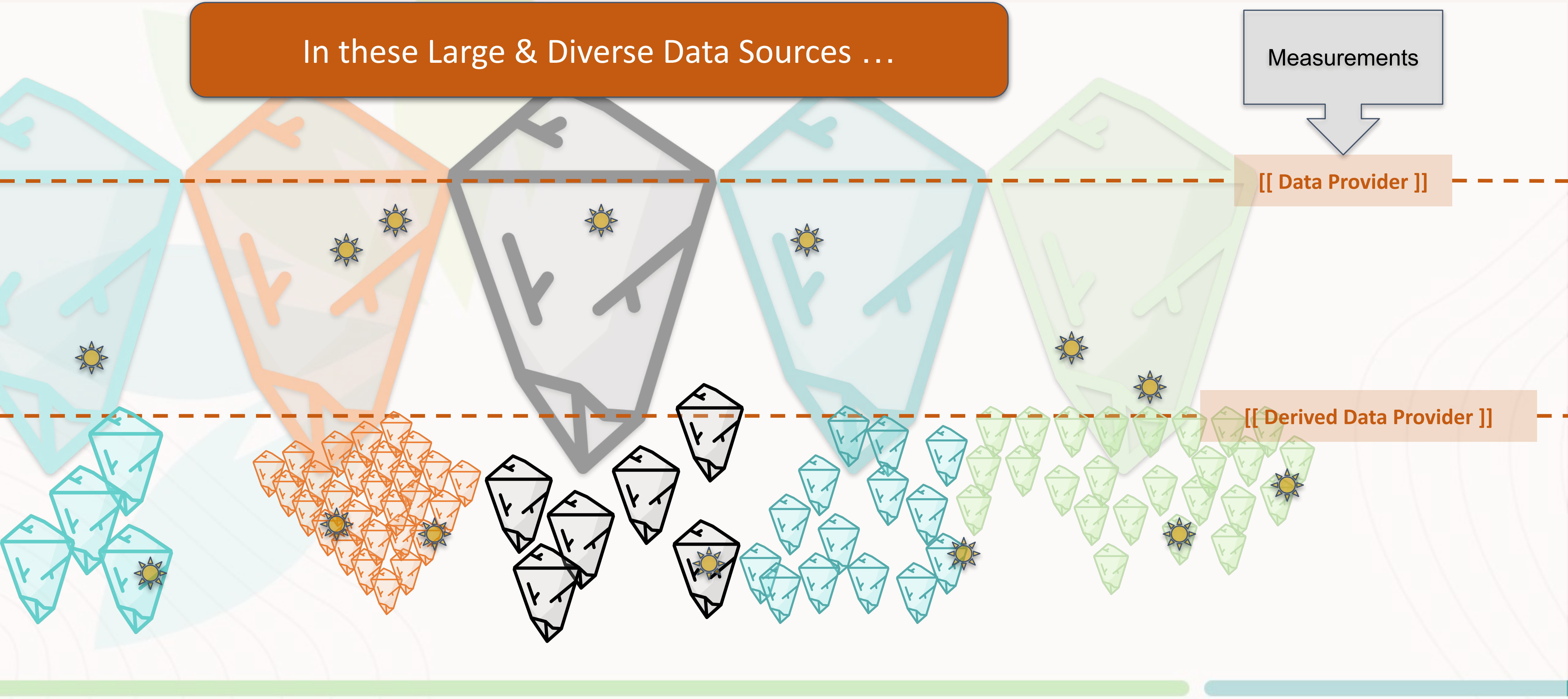
Subset, find and recombine

In these Large & Diverse Data Sources …



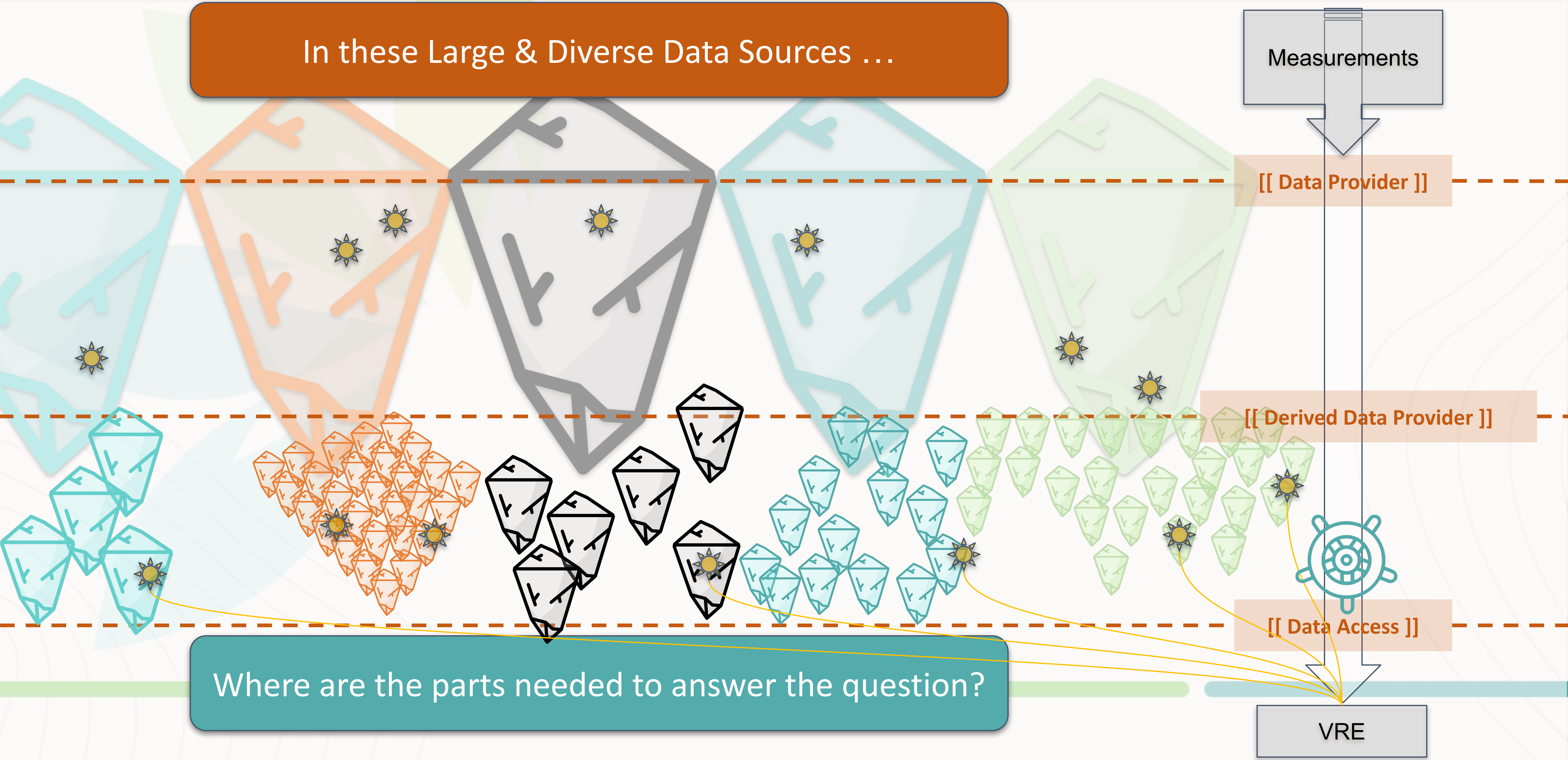
Subset, find and recombine

In these Large & Diverse Data Sources …



Subset, find and recombine

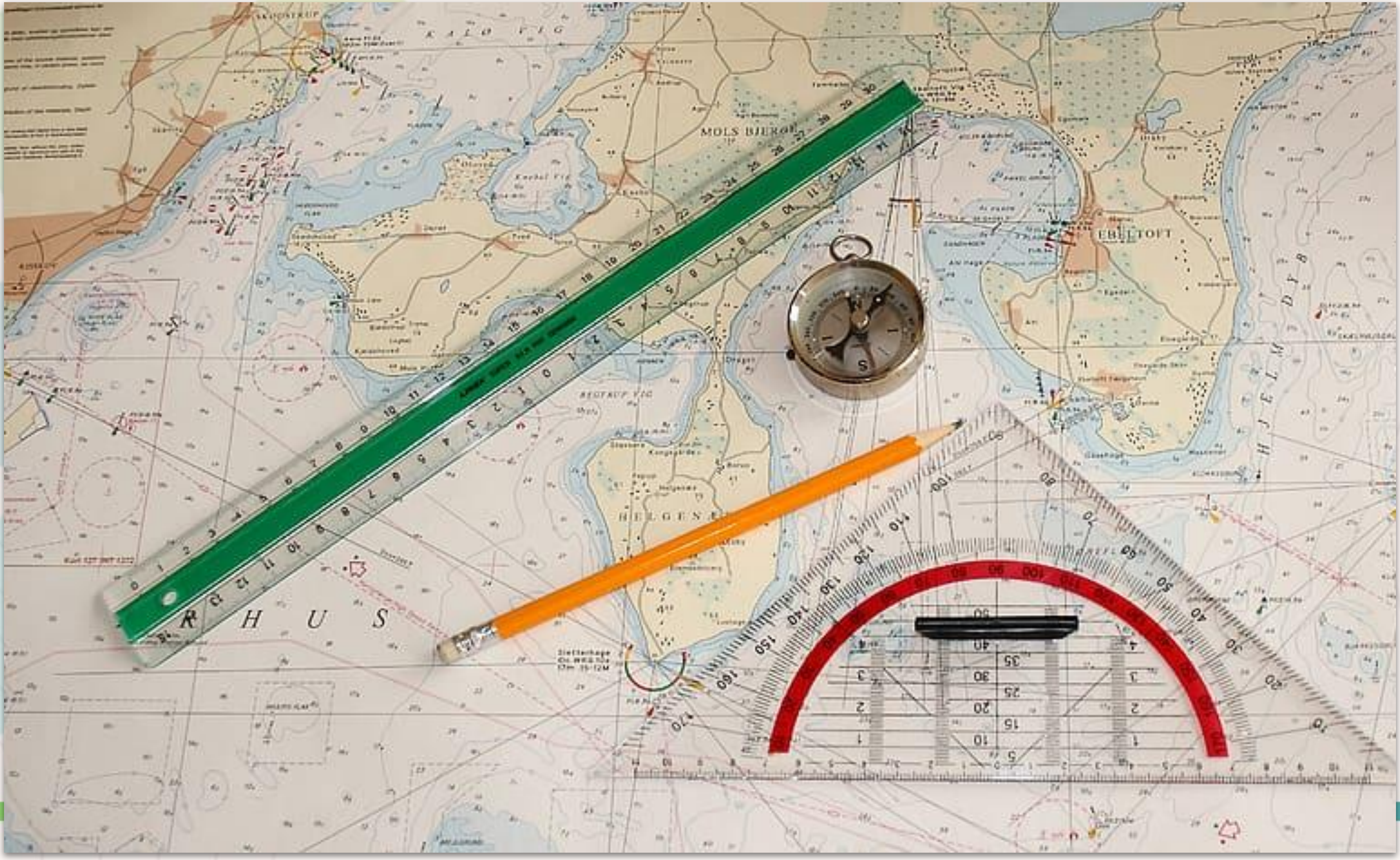
In these Large & Diverse Data Sources …



Where are the parts needed to answer the question?

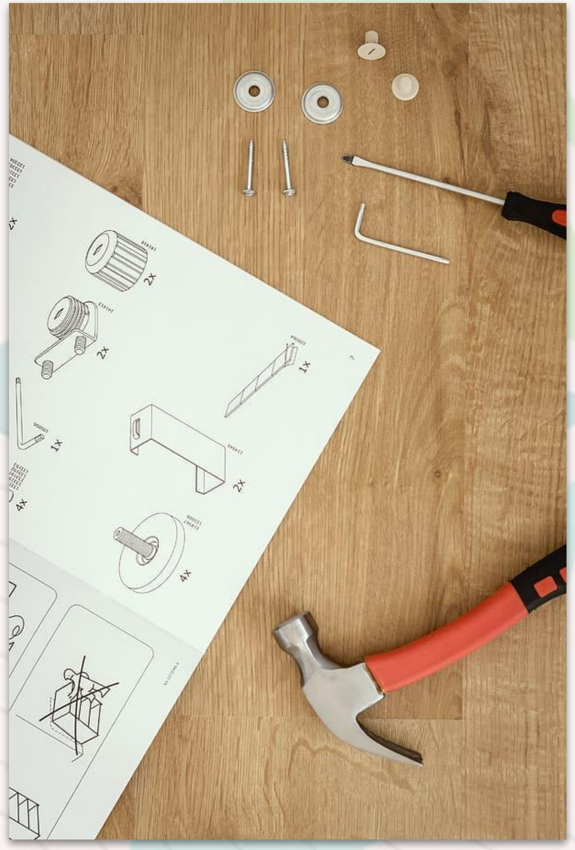
eosc | FAIR-EASE

Navigating the data landscape



eosc | FAIR-EASE

Metadata and semantics



- data = the stuff
- **metadata = the stuff about the stuff**
  - provenance (who, where, when, what, how)
  - citation & attribution
  - general findability (keywords, some links)
- **semantics = stuff about what the stuff is about**
  - making it understandable = interoperable
  - making it machine-actionable
  - is very often overlooked
    - provided for humans only
    - understandable to domain experts only
    - incomplete

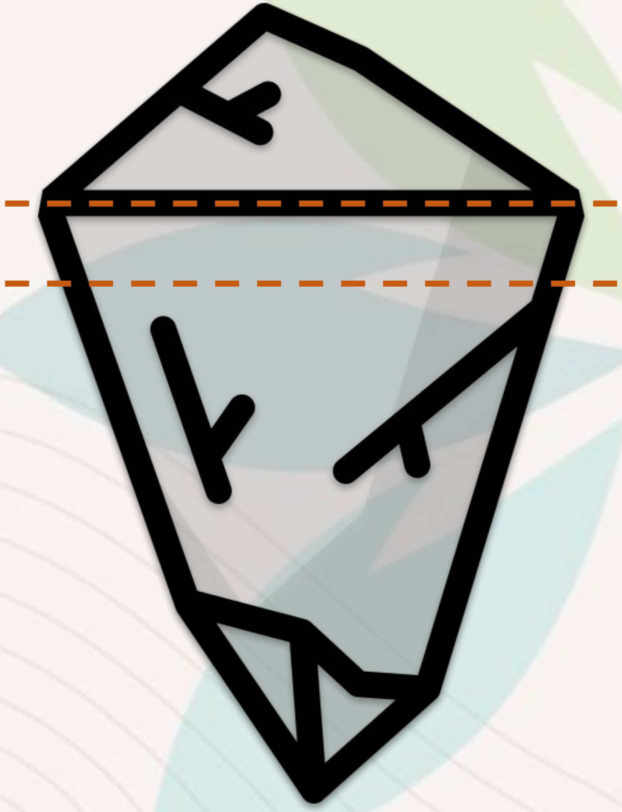
«your new bed»

MALM
299,- €
190 x 210

THE
INSTRUCTION
KIT

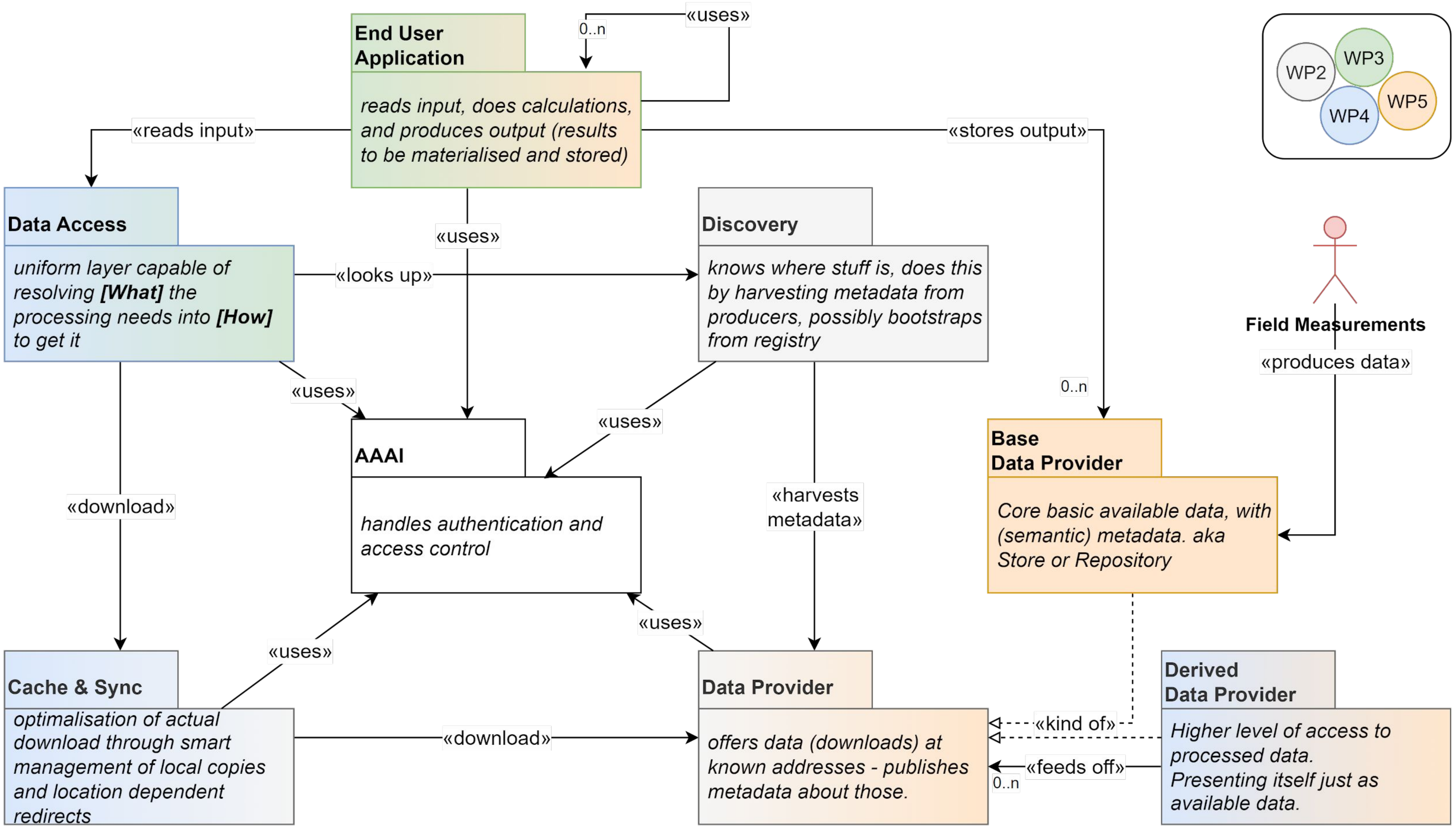
IKEA®

The Metadata iceberg



- the «where» and «when»
    - temporal coverage
    - spatial coverage (both emerging via DCAT)
- the «what» is in there and how is it connected
    - content coverage (not just title and keywords))
    - shapes (rows, columns)
    - histograms / distributions of available values
    - quality coverage
- the «how» for optimal retrieval
    - format
    - size / delivery speed
    - available cache / copies / updates

Separation of concerns



End User Application

reads input, does calculations, and produces output (results to be materialised and stored)

Data Access

*uniform layer capable of resolving **[What]** the processing needs into **[How]** to get it*

Discovery

knows where stuff is, does this by harvesting metadata from producers, possibly bootstraps from registry

AAAI

handles authentication and access control

Base Data Provider

Core basic available data, with (semantic) metadata. aka Store or Repository

Cache & Sync

optimalisation of actual download through smart management of local copies and location dependent redirects

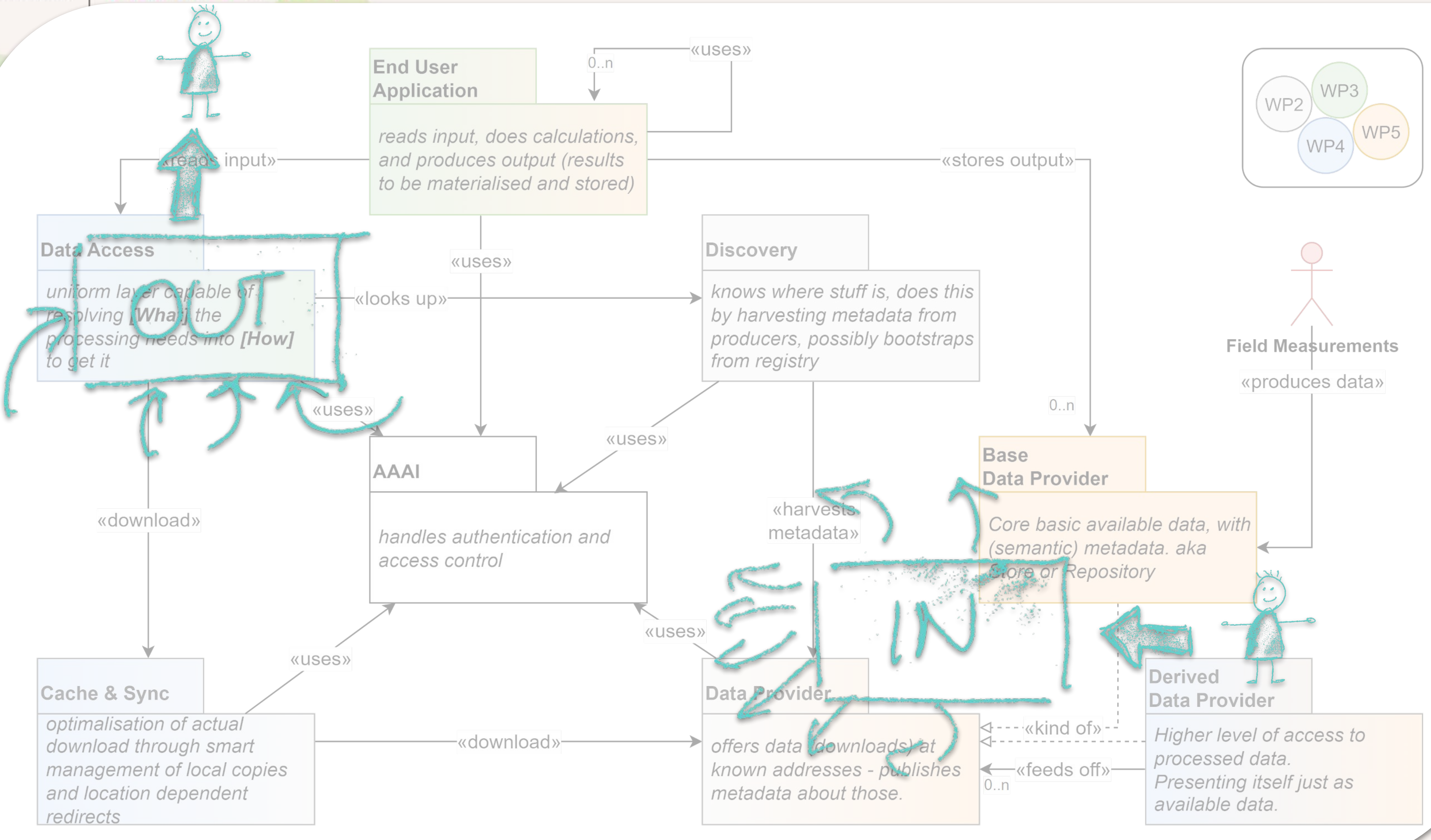
Data Provider

offers data (downloads) at known addresses - publishes metadata about those.

Derived Data Provider

Higher level of access to processed data. Presenting itself just as available data.

I/O view



End User Application
reads input, does calculations, and produces output (results to be materialised and stored)

Data Access
uniform layer capable of resolving [What] the processing needs into [How] to get it

Discovery
knows where stuff is, does this by harvesting metadata from producers, possibly bootstraps from registry

AAAI
handles authentication and access control

Field Measurements

Base Data Provider
Core basic available data, with (semantic) metadata. aka Store or Repository

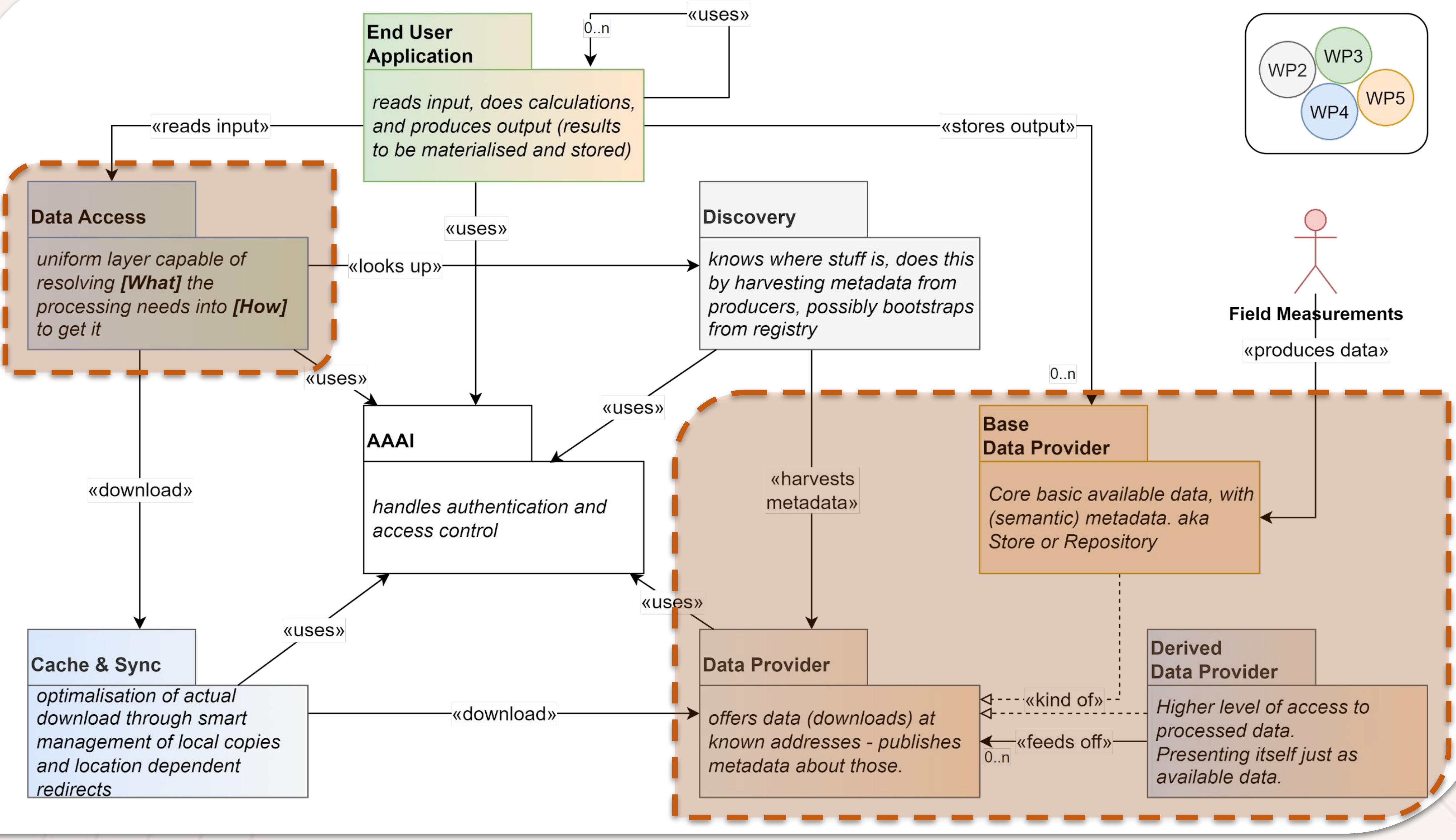
Cache & Sync
optimalisation of actual download through smart management of local copies and location dependent redirects

Data Provider
offers data (downloads) at known addresses - publishes metadata about those.

Derived Data Provider
Higher level of access to processed data. Presenting itself just as available data.

eosc | FAIR-EASE

The next step in our journey



End User Application

reads input, does calculations, and produces output (results to be materialised and stored)

Data Access

*uniform layer capable of resolving **[What]** the processing needs into **[How]** to get it*

Discovery

knows where stuff is, does this by harvesting metadata from producers, possibly bootstraps from registry

AAAI

handles authentication and access control

Base Data Provider

Core basic available data, with (semantic) metadata. aka Store or Repository

Cache & Sync

optimalisation of actual download through smart management of local copies and location dependent redirects

Data Provider

offers data (downloads) at known addresses - publishes metadata about those.

Derived Data Provider

Higher level of access to processed data. Presenting itself just as available data.

Field Measurements

eosc | FAIR-EASE

Questions?

Funded by
the European Union

Data Lake summary

Data Lake will

- Allow data to be *where* they are, *as* they are
- Will be the *specifications* to build an *interoperability layer* over the distributed FE data
- Will allow the DDAS and VRE to plot the routes from the user/client to the requested dataset/data subset/combined data
- Will allow data (sub)sets to be identified and retrieved via URLs, built on-the-fly
- Will have its work cut out for it in figuring out
  - how to obtain all the machine-actionable metadata to allow data to be fully understood
  - how to subset data in an efficient way