

UNIVERSITY OF TWENTE.



A PROCESS-BASED FRAMEWORK TO ENABLE THE TRANSITION FROM GEOSPATIAL WEB SERVICES TO GEOSPATIAL WEB APPLICATIONS



JAVIER MORALES & ROLF A. DE BY



FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION



MOTIVATION

- Need for a *formal design theory* for SDI and its components
- SDI development for a *diverse customer base*

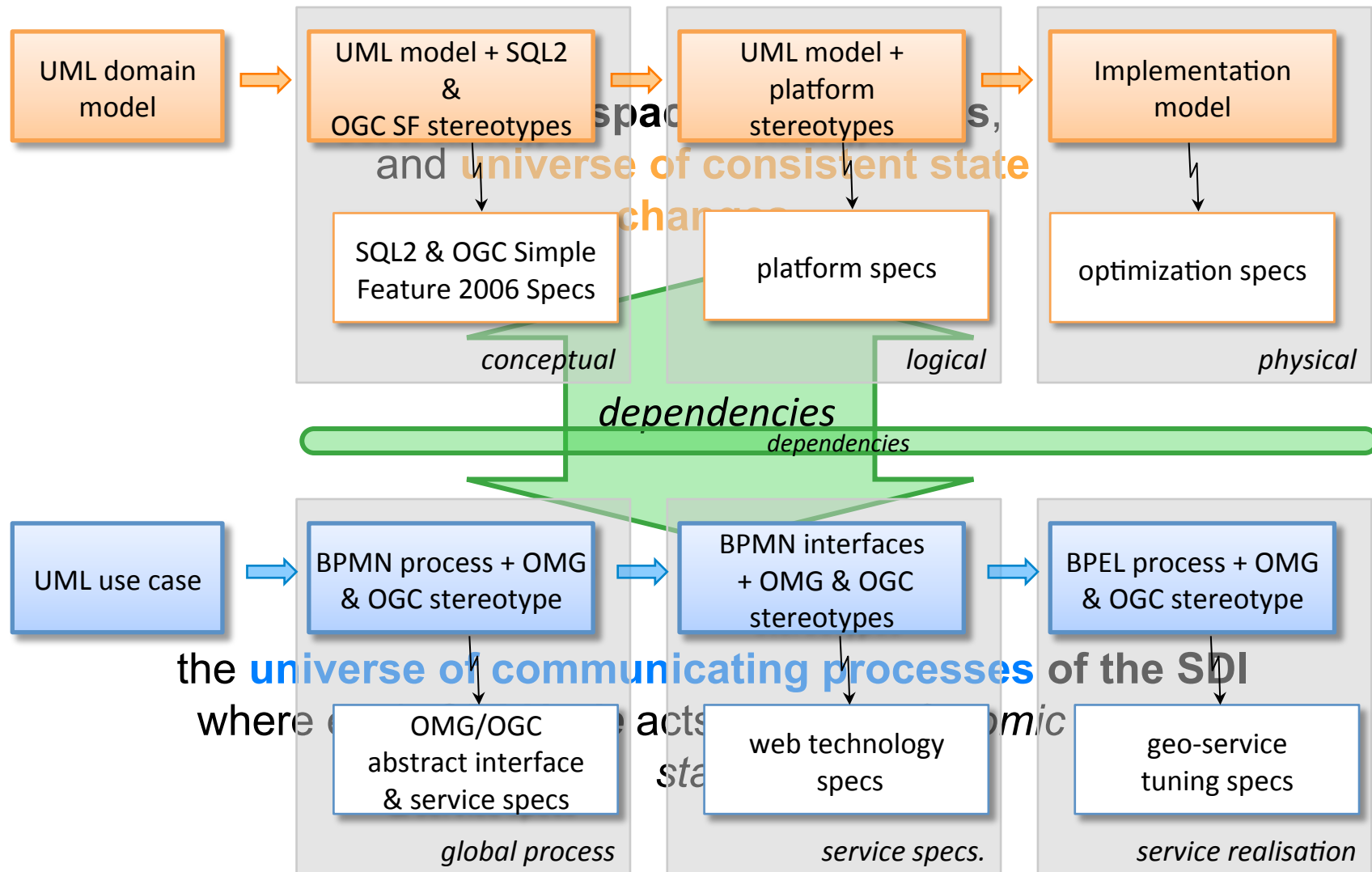
- SDI nodes are *complex IT systems*, with long lifetime expectancy
- Together they form overall *institutional, hybrid* or *ad hoc* SDI
- True proof-of-the-pudding is in the *orchestration* of *across-platform geo-processing workflows*.

- Their design deserves an abstract, platform- & standards-independent definition combined with precise documentation
- Need for abstraction, specification, formal semantics, proofs of properties of
 - *nodes and infrastructure, and*
 - *their collaborative operation*

REASONING

- to improve **quality** and **rigour** of whole development process
- to improve **integrity**, **reliability** or other characteristics of the system
- to improve **requirements** definition
- to provide a firmer foundation for maintenance and enhancement
- to improve **documentation** and **understanding of designs**
- to meet particular **customer** or **standards requirements**
- to explore the properties and **possibilities** of a design architecture

SNEAK PREVIEW



CHALLENGES

SDI node construction is naturally **state-based**

- Which object types?
- Which data sources?
- Which interrelationships?
- Which object life cycles?

Challenging clash

We are introducing

- Rigour,
- Robustness, and
- Replicability

SDI construction is naturally **process-based** in this process

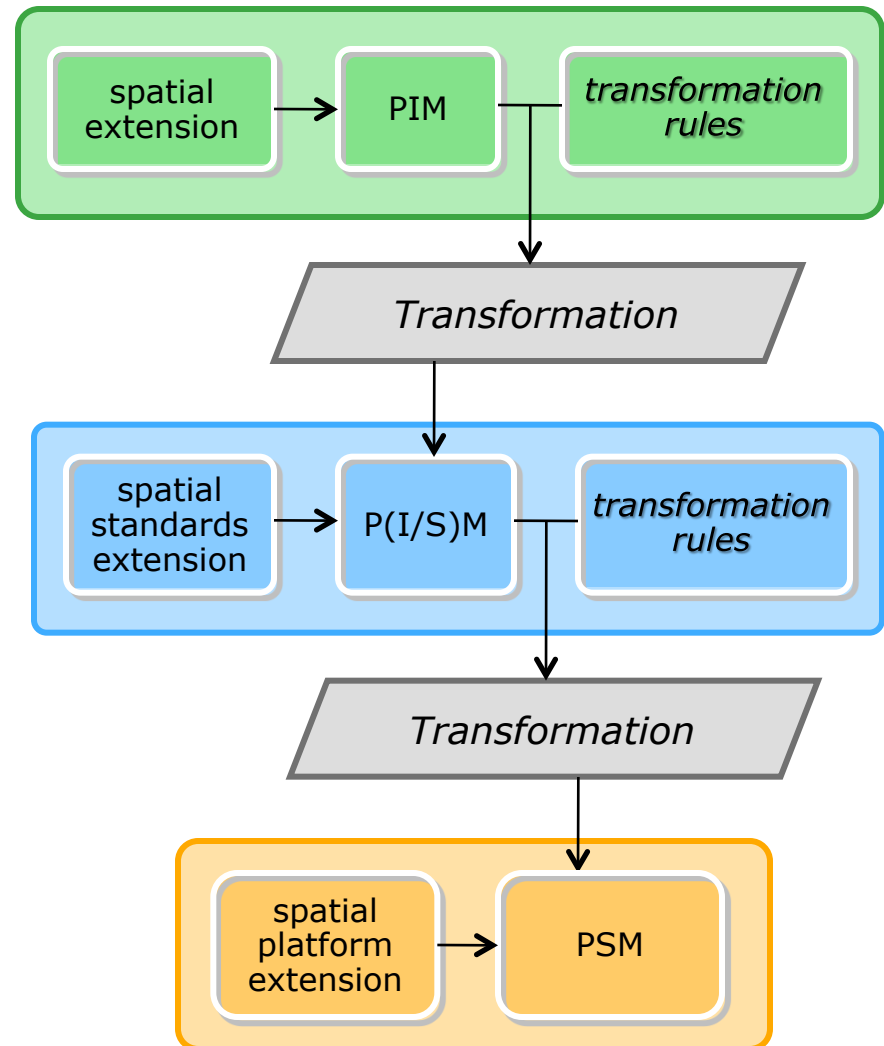
- Which communicating processes?
- Which are the needed synchronizations, and which are the allowed asynchronies?
- Which information exchanges must or can occur?
- Which interruptions do we allow?
- What are the service levels?

CREDO & STARTING POINTS

- Ref. Model for Open Distributed Processing (**RM-ODP**)
 - Methodological baseline for ISO 19100 series and OpenGIS RM
- *Separation of design concerns:*
 - information content
 - logical structuring
 - physical organisation
 - process structure
 - standards compliance
 - data synchs & QoS levels
- *Transformational design approach*
- Use of restricted UML:
 - the state and process parts that are formalized or formalizable
 - with a geomatic extension based on accepted (ISO/OGC) standards

TRANSFORMATION PROCESS → Geo-MDA

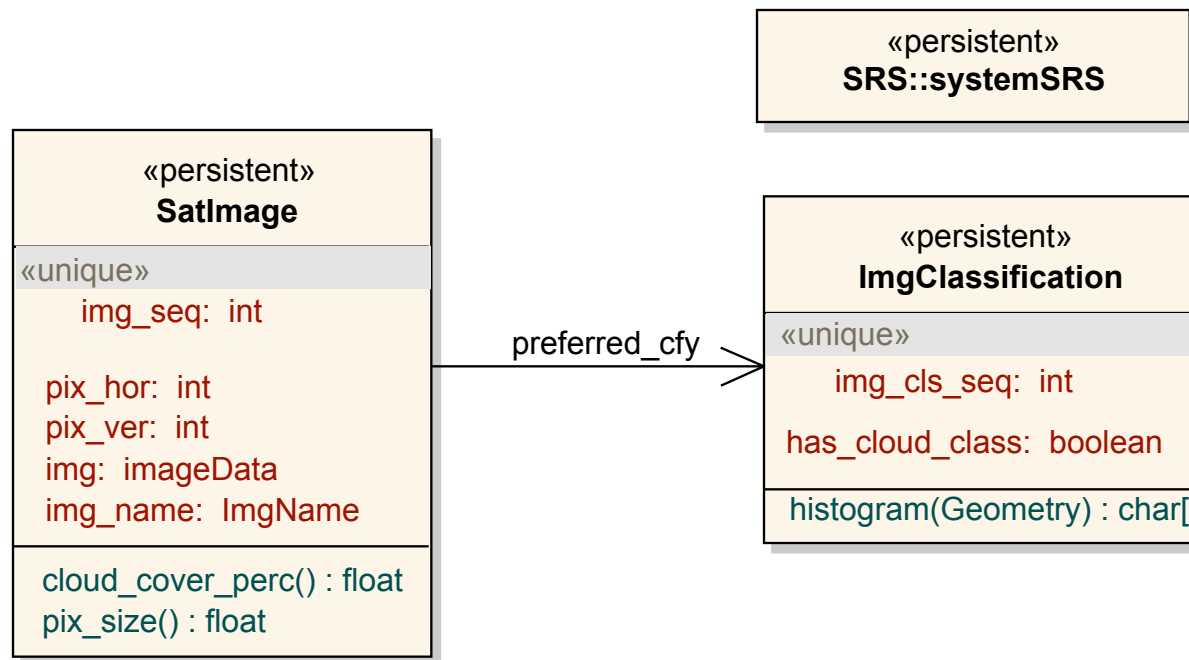
- 'Spatially enable' the design method
- Account for abstract specs and implementation specs
- Allow for choices of implementation platform



AN EXAMPLE CASE SPECIFIED

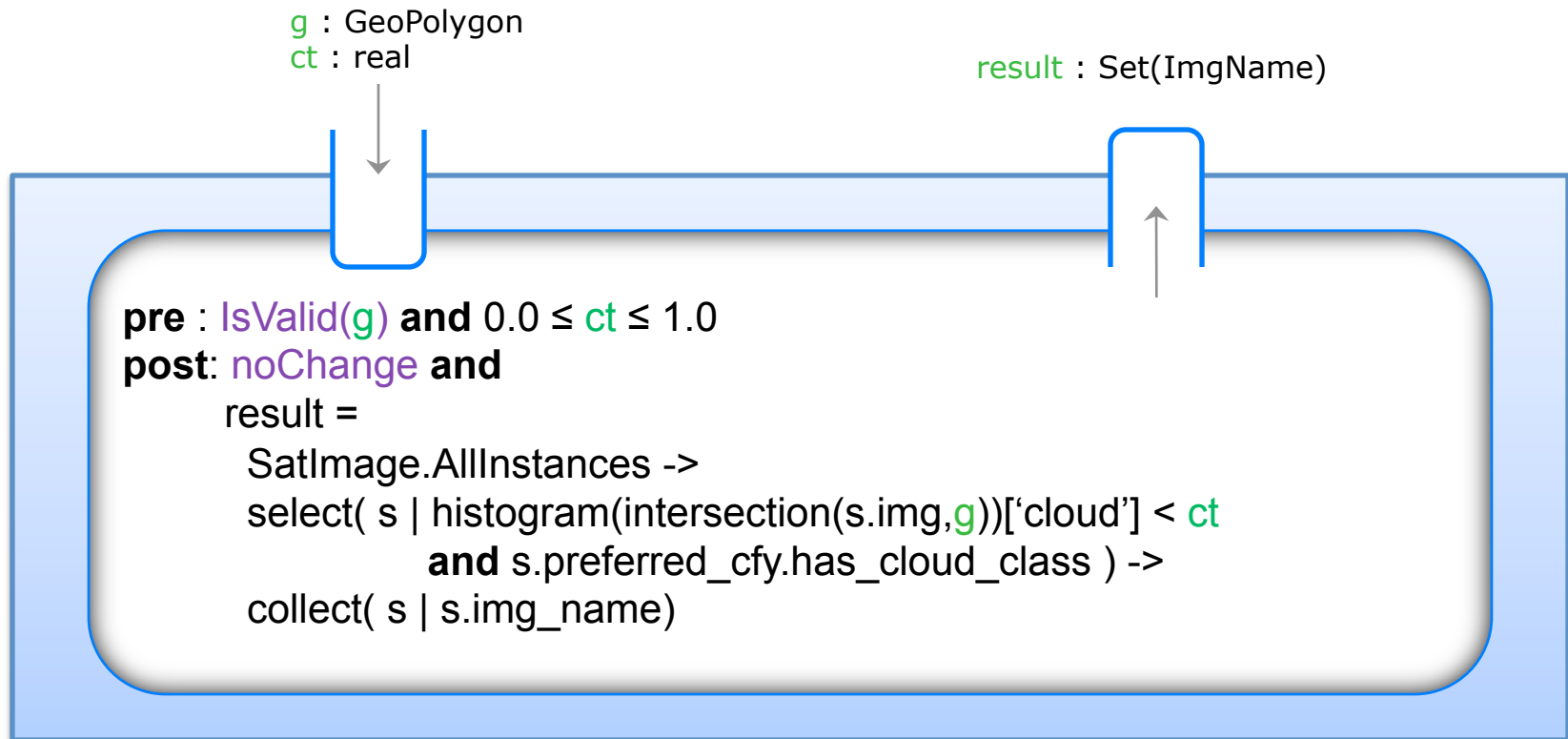
- The regional RS Agency is the central archive for specific sources of satellite data.
- One of its services to clients is the provision of classes of satellite image, *restricted to some arbitrary geographic extent*.
- In this example, we specify the service to *list available images* that meet the quality characteristics of *limited cloud cover* within the area requested.

CONCEPTUAL INFORMATION CONTENT



Observe: Conceptually a simple, partial data model
In standard, but restricted UML,
including *OGC data standards*, *OGC-enriched OCL*.

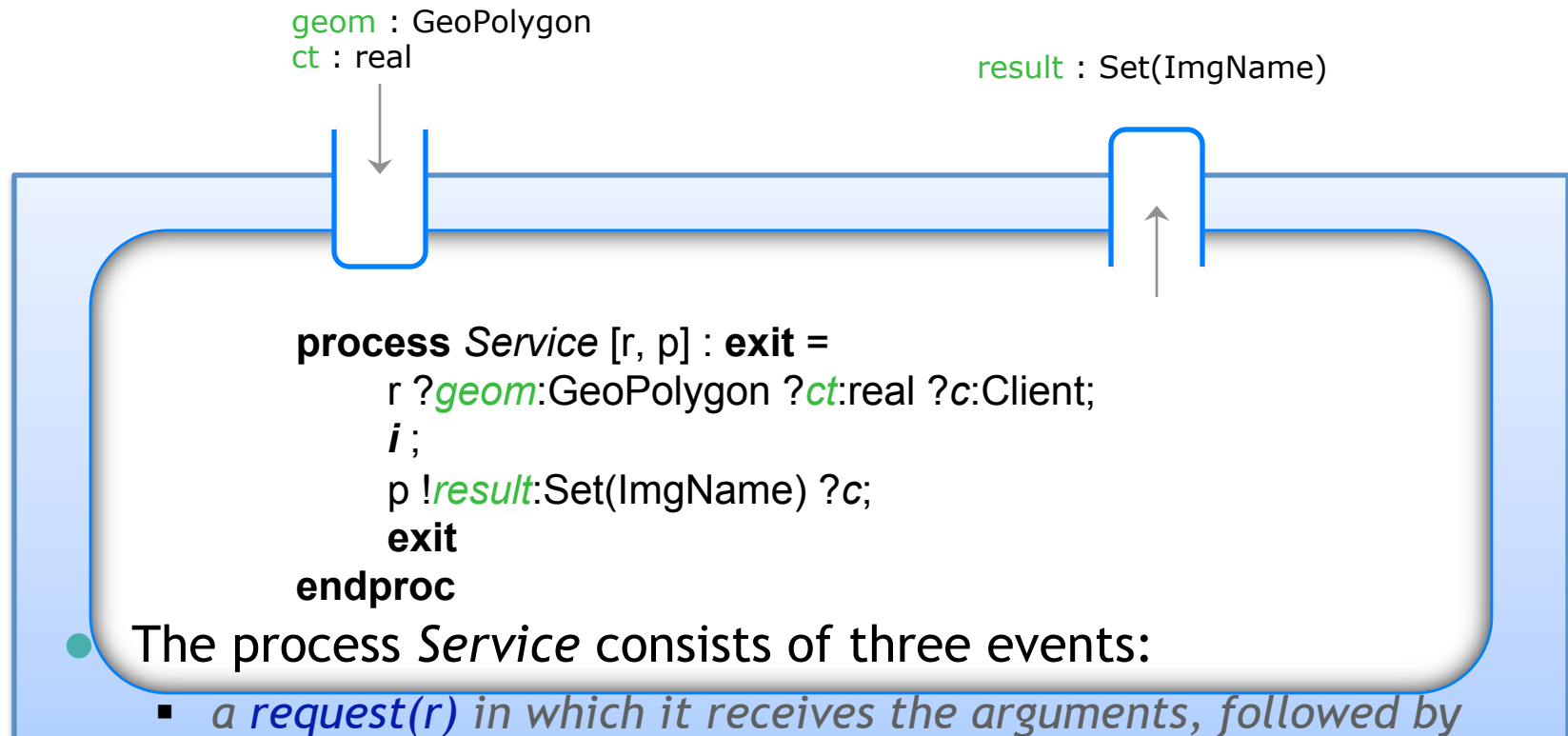
A CONCEPTUAL SERVICE OFFERING



Observe: Conceptually a single service

THE SERVICE OFFERING

IN PROCESS-ALGEBRAIC EVENT TERMS



THE SERVICE OFFERING

IN PROCESS-ALGEBRAIC EVENT TERMS

- The client process

```
process Client [r, p] : exit =  
  r !geom:GeoPolygon !ct:real !c:Client;  
  p ?result:Set(ImgName) !c;  
  exit  
endproc
```

- Our SDI-mediator as a synchronised process composition:

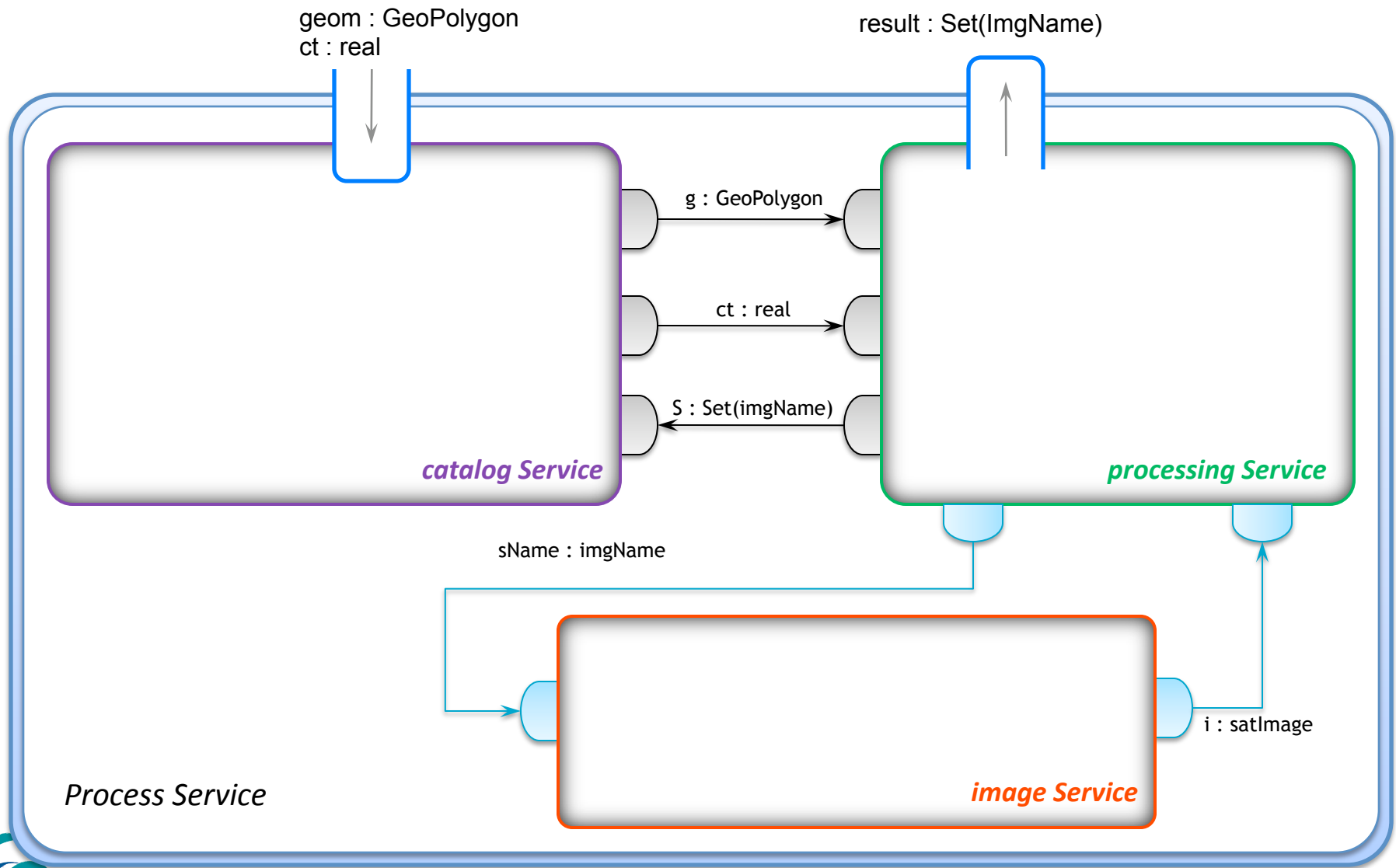
```
behavior mini_SDI =  
  Service [r,p] || Client [r,p]  
endbehavior
```

- Service and Client synchronize on all events common to them

PROCESS & EVENT SPECIFICATIONS

- event sequencing ;
 - full interleaving |||
 - partial synchronisation |[...]|
 - full synchronisation ||
 - process enabling >>
 - process disabling [>
 - event hiding **hide** ...
 - event guarding [*guard*]
-
- Process algebras are great tools for the precise and verifiable description of service orchestrations

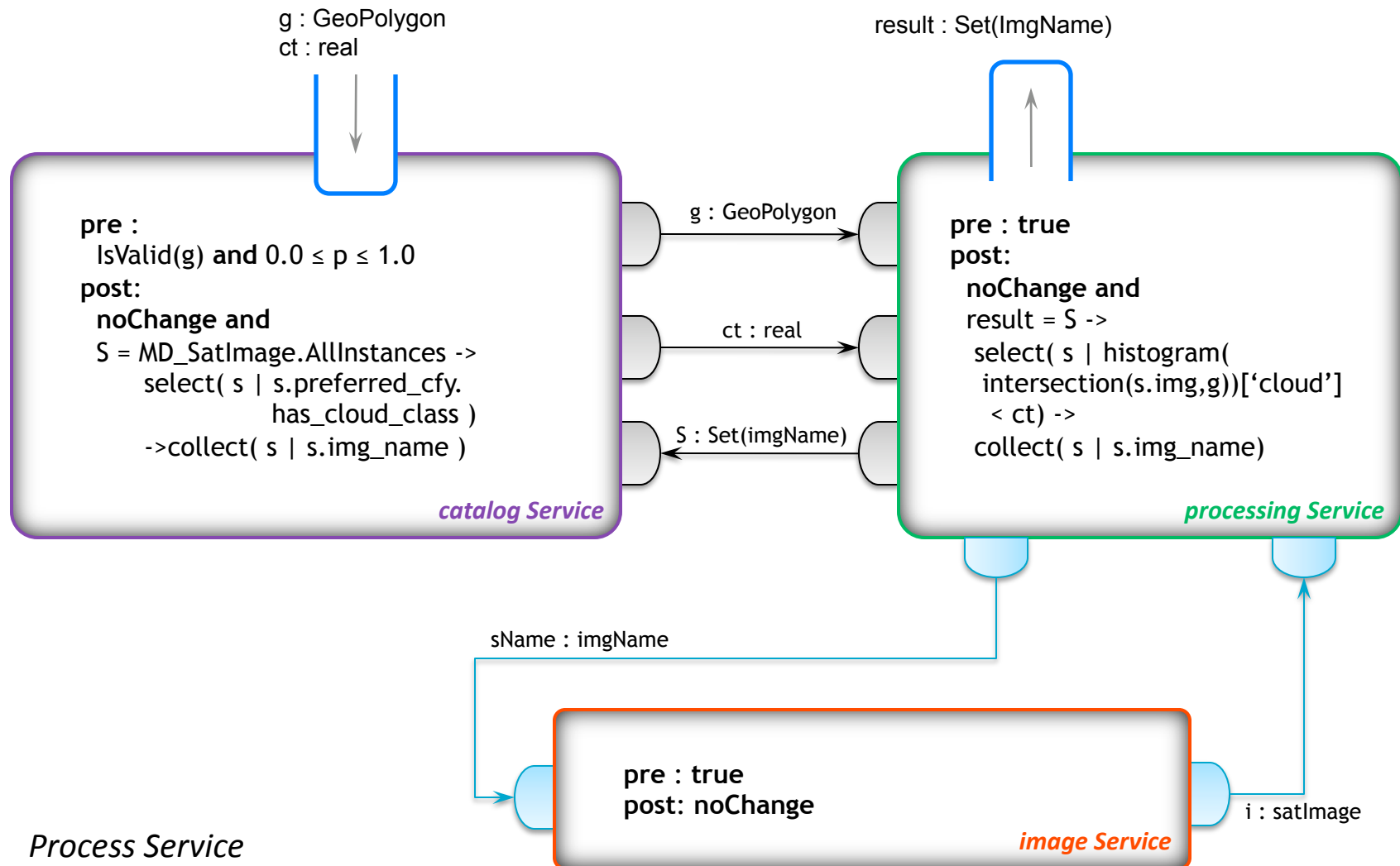
THE SAME SERVICE LOGICALLY DECOMPOSED



THE REFINED PROCESSES

- **process** *Catalog* [r, q] : **exit** =
 r ?geom:GeoPolygon ?ct:real ?c:Client;
 q !geom !p !c !S:Set(ImgName);
 exit
endproc
- **process** *Processor* [r, p] : **exit** =
 q ?geom:GeoPolygon ?p:real ?c:Client ?S:Set(ImgName);
 for *sname* ∈ S: s !sname:ImgName ?i:SatImage;
 i ;
 p ?result:Set(ImageName) !c;
 exit
endproc
- **process** *Imager* [r, p] : **exit** =
 s ?sname:ImgName !i:SatImage;
 Imager [r, p] [] **exit**;
endproc

THE REQUIRED SERVICE LOGICALLY DECOMPOSED

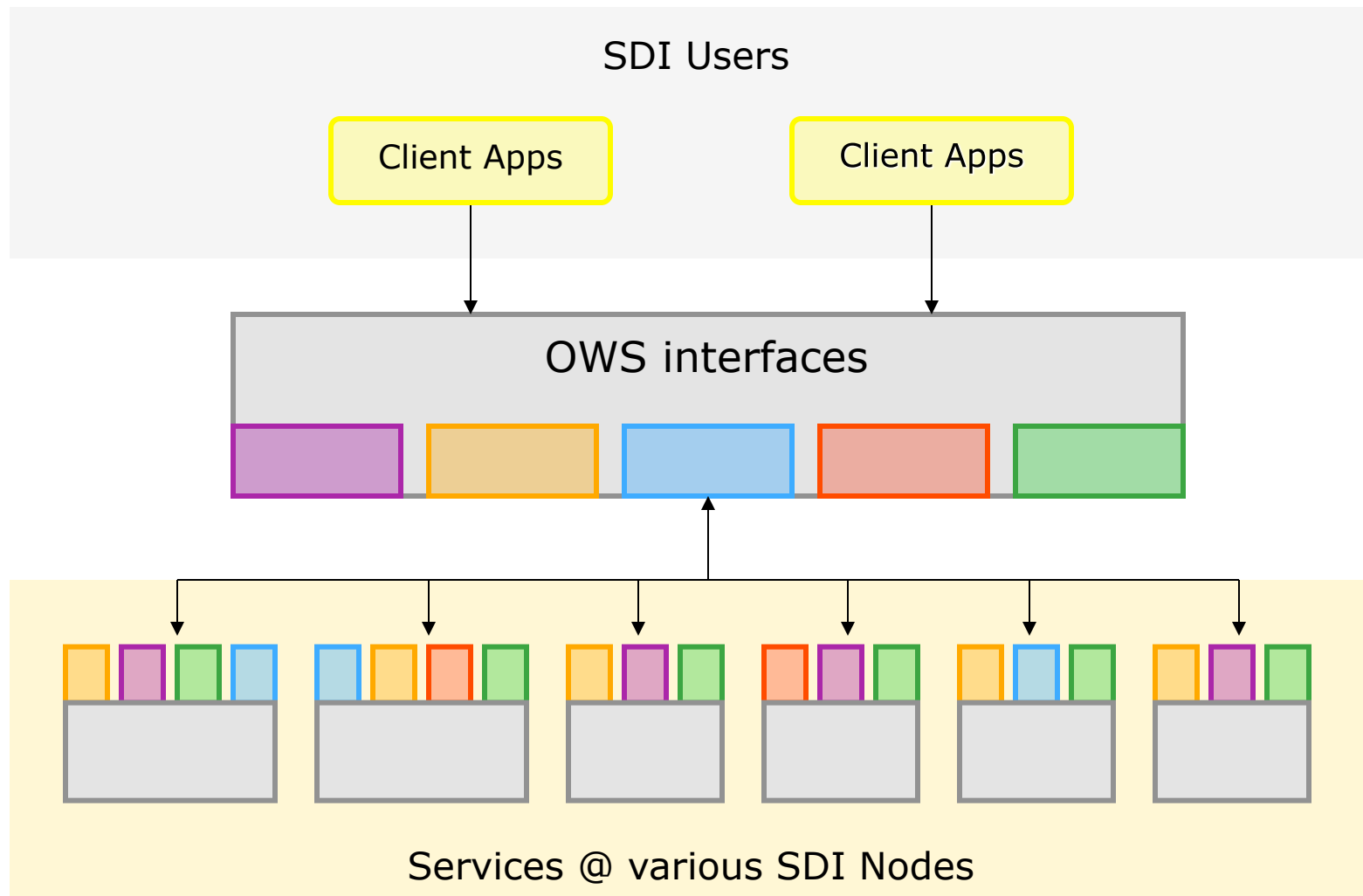


Process Service

image Service

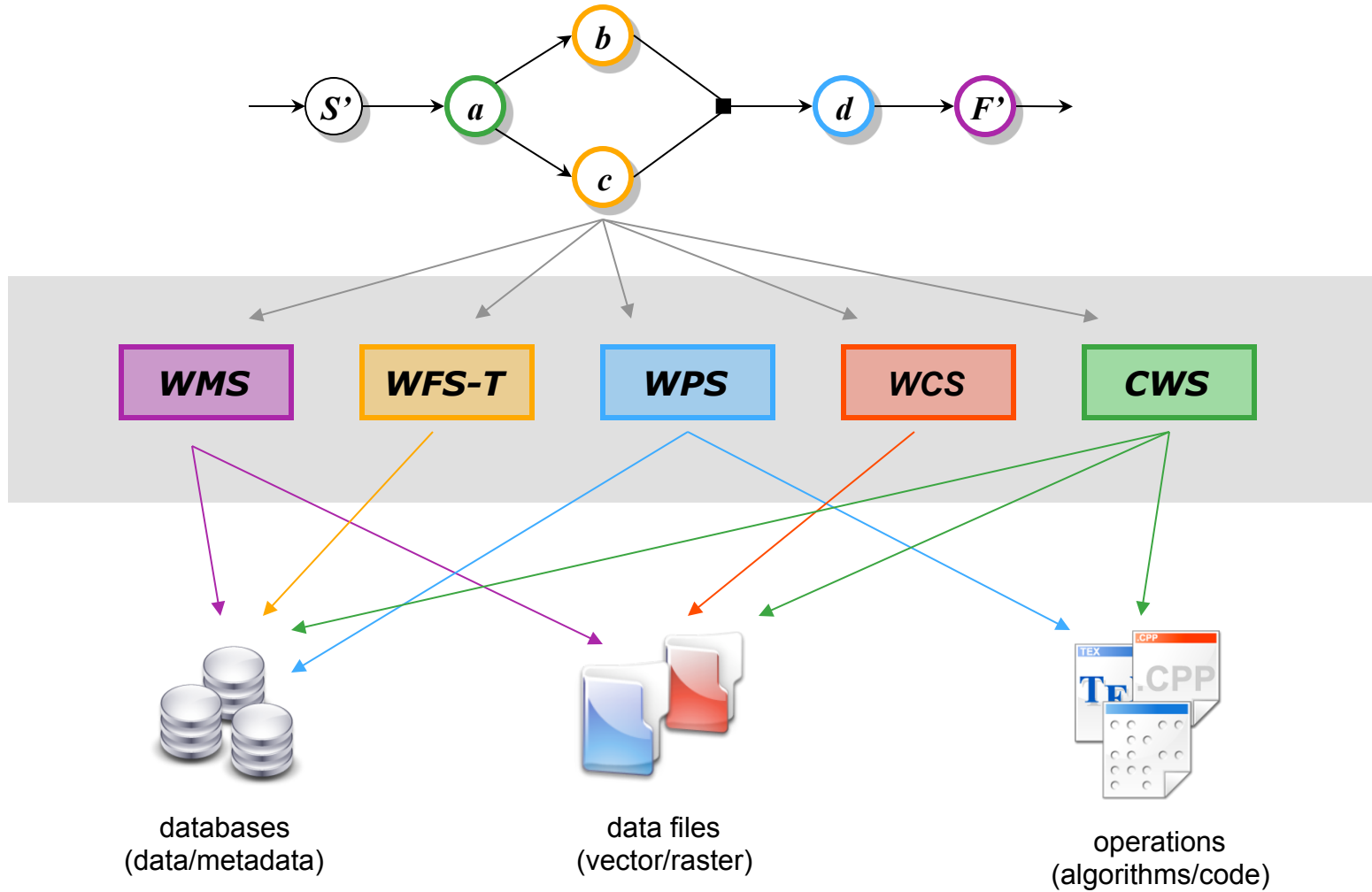
SERVICE MAPPINGS

TO ATOMIC SERVICE OFFERINGS AVAILABLE IN THE INFRASTRUCTURE



THE SERVICE REALISATION

The decomposed workflow (according to the specification)

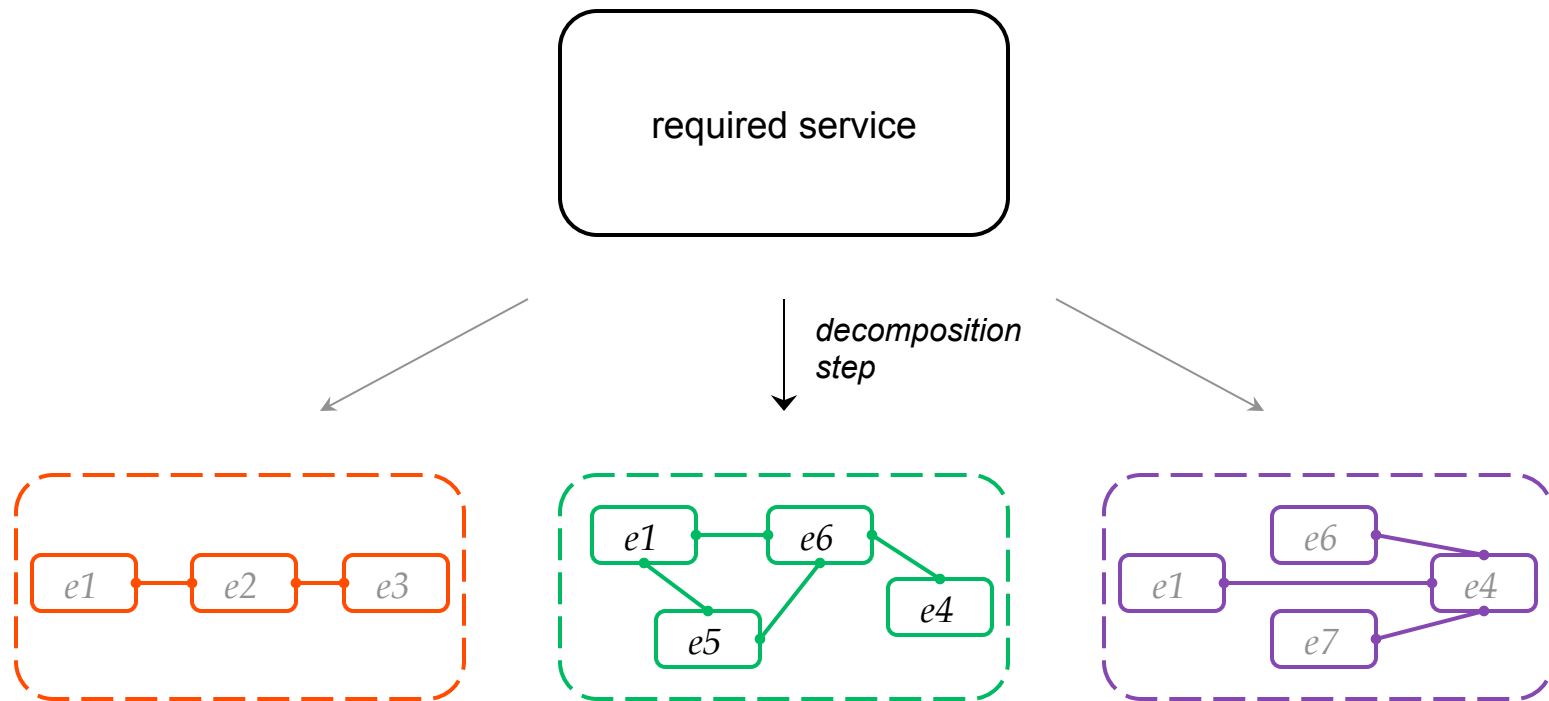


THE SERVICE REALISATION

- Look at QoS issues, depending on server capabilities:
 - Use size of img, img's cloud cover percentage and size area of interest g to selectively filter out potentially useful images.
 - Perform true image computations possibly in parallel on separate compute servers
 - Use bbox(g) to selectively retrieve image data only
 - *In short:* apply deeper semantic understanding to improve the process's efficiency

ALTERNATIVE SERVICE DECOMPOSITIONS

A DESIGNER'S CHOICE (*AVAILABILITY, QUALITY, PERFORMANCE, COST, etc.*)



FINALE

- This work attempts to build up methodical standards for formal documentation of complex SDI (nodes)
- An SDI node is a *responsive state manager*, an SDI is a large *communicating process*
- State-based and process-based transformational design is key, and should describe the platform technology- & standards-independent intrinsics of node(s) and SDI.
- Challenges: many!
 - full-proof and fool-proof transformations
 - human capacity
 - case experience & fitness-for-use metrics
 - Inclusion of deeper semantics

FINALE

OUR AIMS

- build a body of design & implementation knowledge for SDI nodes
- shared philosophy of modular system architecture & SOA
- standardized methods of design & implementation
- share in solutions over a collaborative knowledge network for SDI technology



A SOLUTION WITHOUT A FORMAL SPECIFICATION

THEY MAY NOT BE WHAT YOU EXPECT



Environmentally-friendly car
based on an alternative
source of energy



TYPICAL PROBLEMS FOR NOW AND FOR THE FUTURE

- How do we formalize user requirements?
- Does an SDI node implement its specs?
- Is the offered service fit for its purpose?
- Does the design conform to agreed standards?
- How and to what extent do we achieve specification independence from choice of standards and platforms?
- Will a service orchestration give the required results?
- Do functional equivalents exist with better efficiency characteristics?
- How can the specification contribute to auto-generation of metadata?