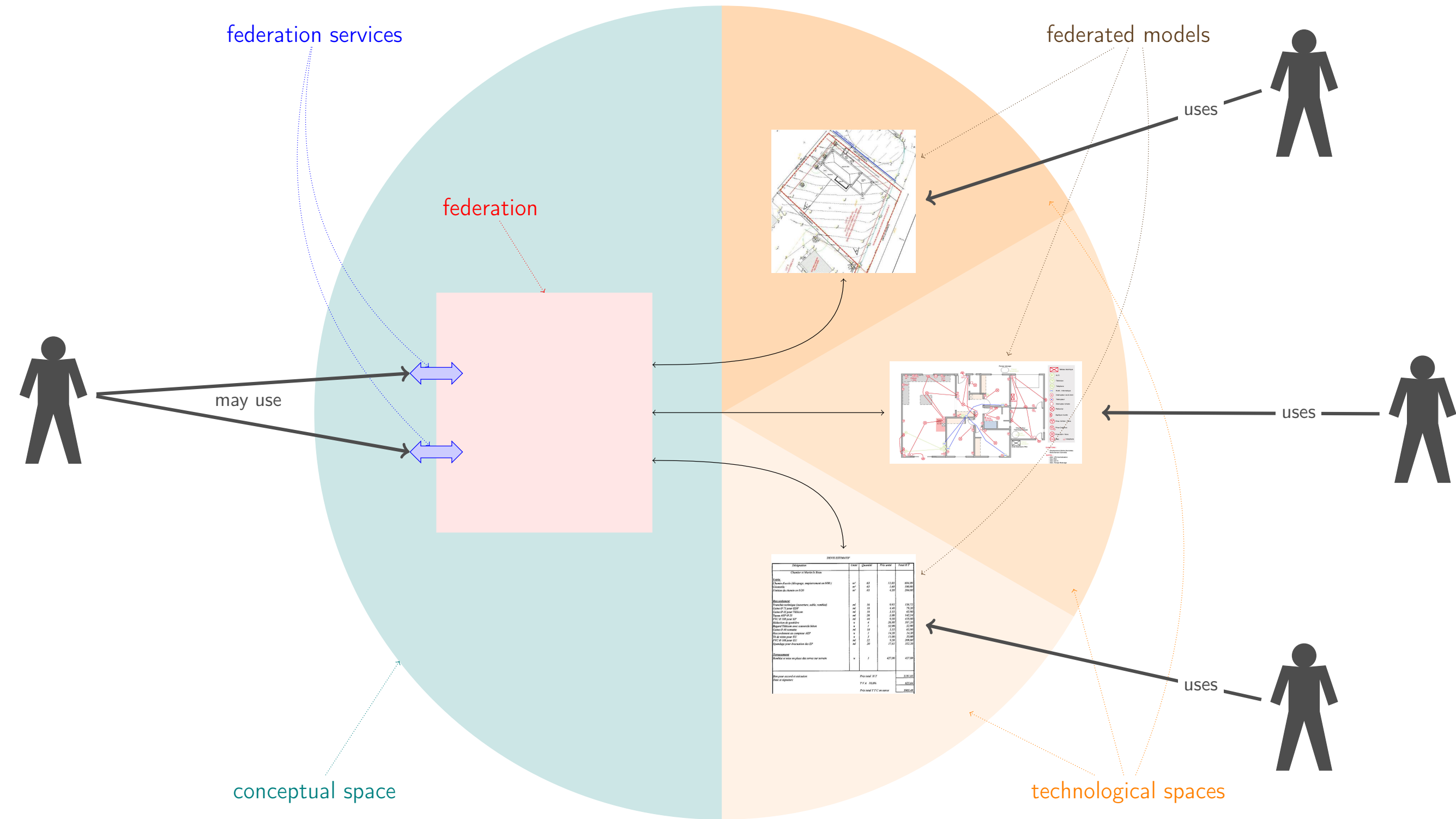


Why model federation?

- Working on complex systems requires to operate simultaneously on several points of view
- A point of view corresponds to specific practices in terms of processes, teams, tools...
- These views are interdependent, one element of the system may be part of several views
- ⇒ We have to manage their consistency / inconsistency without disrupting existing practices



What is model federation?

- Federated models remain in their original technological spaces, users keep their practices
- Their correspondences are reified within a model of the conceptual space called a federation
- A federation is connected to a set of federated models and manages their correspondences
- A federation can provide new services

Model federation with Openflexo

- Openflexo is an open source software framework supporting model federation
- It relies on a DSML for model federation: FML (Federation Modeling Language)
- Openflexo features an Integrated Development Environment and an interpreter for FML
- It includes libraries to connect to federated models called technology adapters
- The main technology adapters provided are EMF, UML, OWL, Word, Excel, XML, JDBC, PDF

FML

- FML structures a federation using virtual models that contain concepts
- It is a full-fledge typed executable object oriented modeling language
- Concepts (≈ classes) feature properties (≈ attributes) and behaviors (≈ methods)
- Properties may refer to federated models using model slots defined by technology adapters
- Behaviors define in an imperative language how concept instances are managed
- FML supports abstraction, encapsulation, multiple inheritance, polymorphism, reflection...

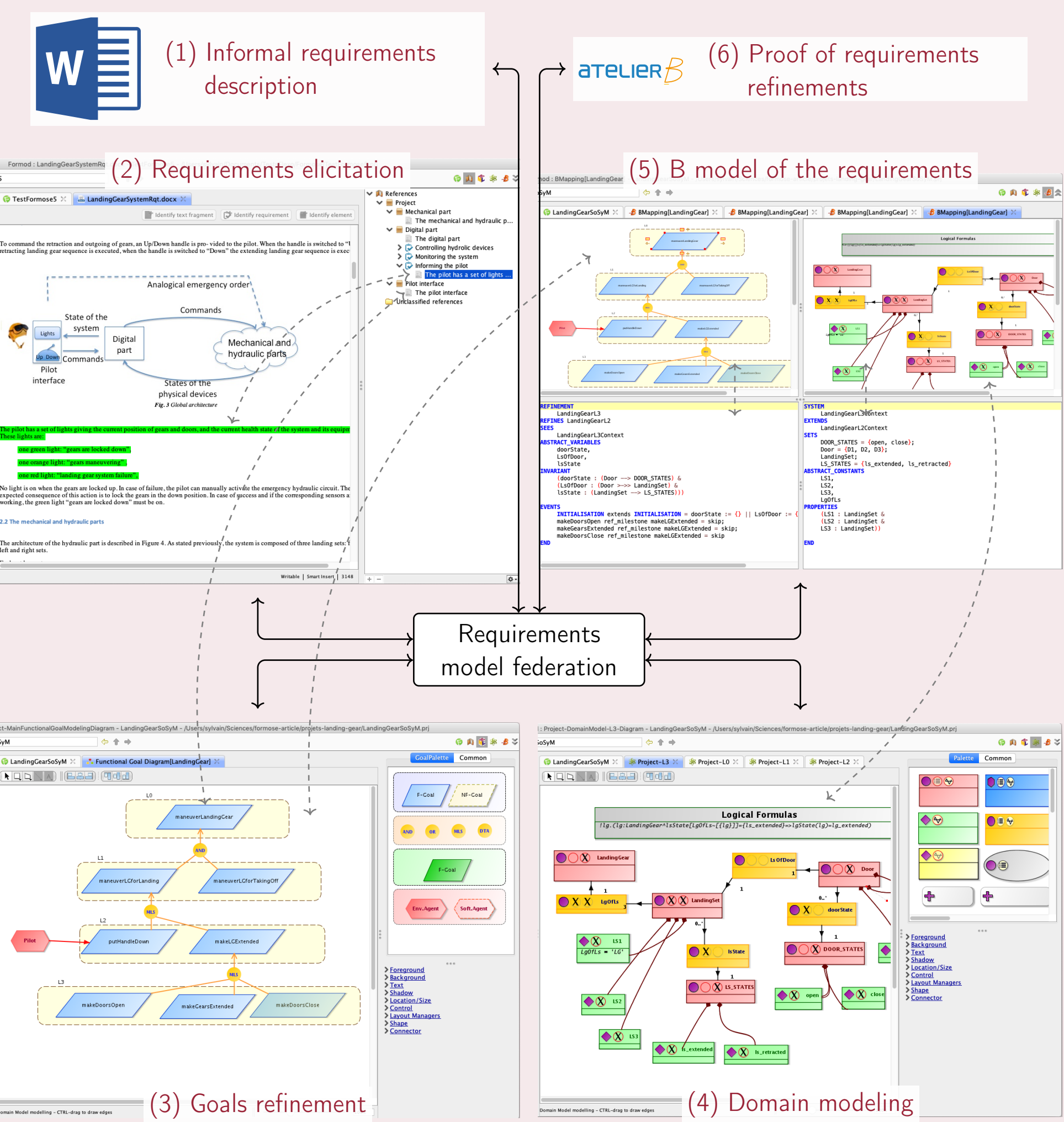
```

namespace "http://www.openflexo.org/house/" as HOUSE;
use org.openflexo.technologyadapter.excel.SemanticsExcelModelSlot as EXCEL;
use org.openflexo.technologyadapter.dxf.DXFModelSlot as DXF;
@URI(HOUSE + "BuildingSite.fml")
@Author("Fabien Dagnat")
public model BuildingSite {
  SitePlan site with DXF::DXFModelSlot();
  EstimateModel estimate with EXCEL::SemanticsExcelModelSlot(mappedModel=EstimateModel);
  public concept Land {
    Area area with ConceptInstance(container=site);
    ServicingCost servicingCost with ConceptInstance(container=estimate);
    create(Area area, ServicingCost sc) {
      this.area = area;
      this.servicingCost = sc;
    }
  }
  void estimate() {
    this.servicingCost.setAreaSize(this.area.getSize());
  }
}
    
```

3 application examples

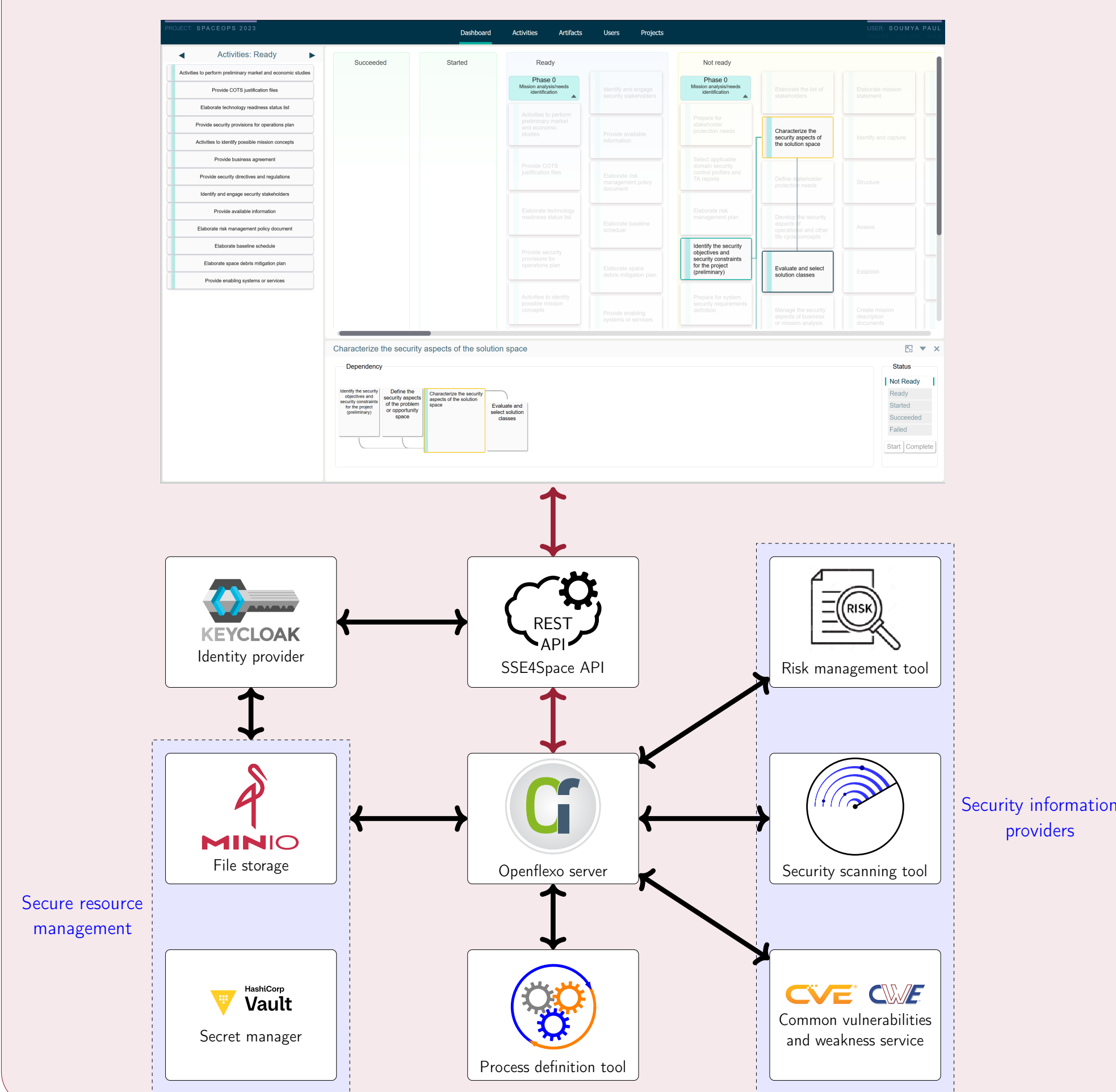
Formod

- A model-based requirements engineering tool
- Federation containing 19 VM, 177 concepts (≈ 8000 sloc)



SSE4Space

- Managing the security concern in satellite development
- Federates security information with process information
- Developed with Telindus for European Space Agency



Free modeling

- Tool to co-construct a model and its metamodel
- Centered around an example diagram that may not conform to a metamodel
- Provides actions on the model, its metamodel and their representations

