LangDev CON 2024

Open Science principles in software product lines: The case of the UVL ecosystem

David Romero-Organvidez, José A. Galindo, Megha Bhushan, Jose-Miguel Horcas, David Benavides

Open Science principles in software product lines: The case of the UVL ecosystem

David Romero-Organvidez, José A. Galindo, Megha Bhushan, Jose-Miguel Horcas, David Benavides

1. Variability and Software Product Lines
2. Yet another language: UVL
3. UVLHub and Open Science
4. flamapy
5. Conclusions

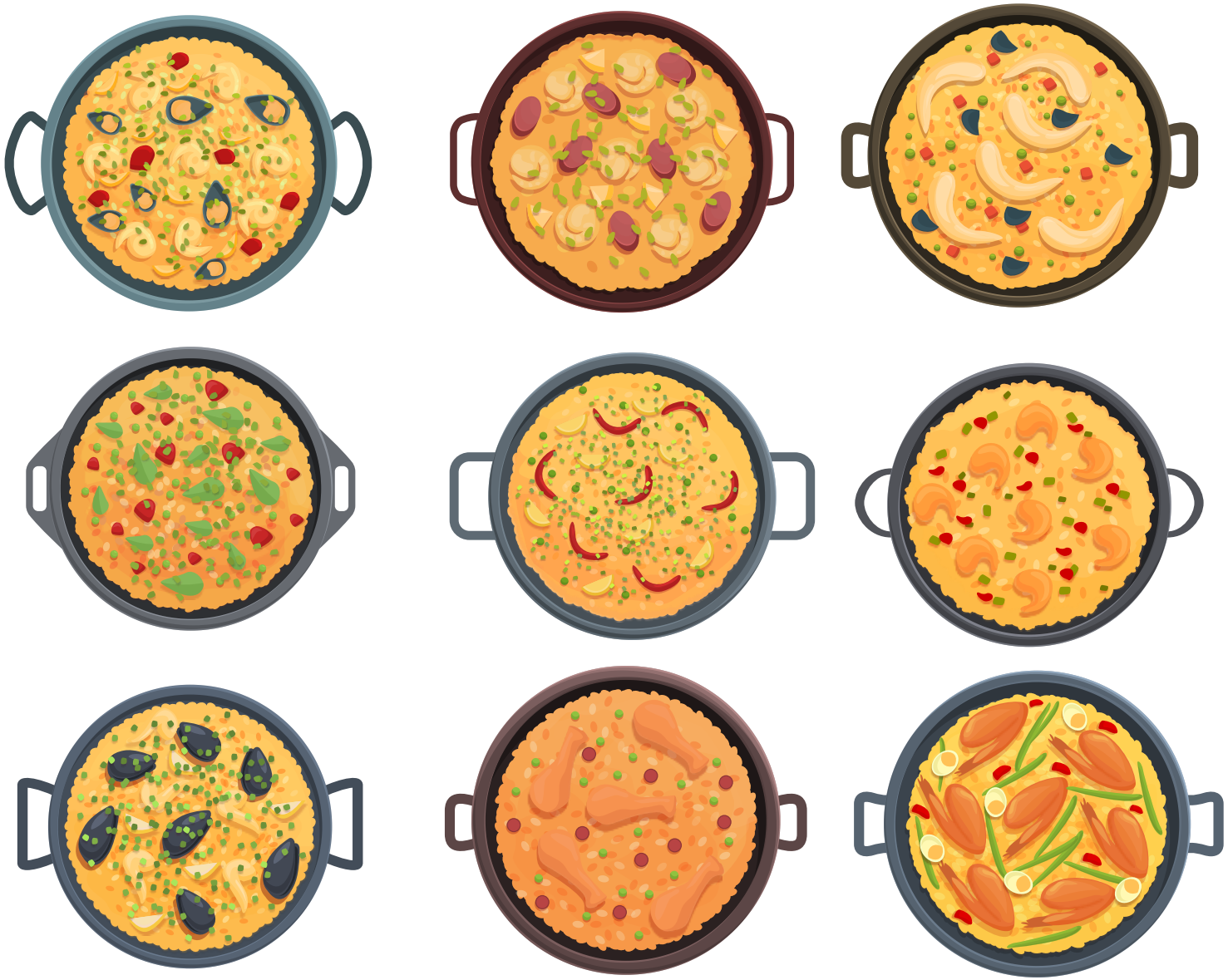
1. **Variability and Software Product Lines**
2. Yet another language: UVL
3. UVLHub and Open Science
4. flamapy
5. Conclusions

1. Variability and Software Product Lines

Software Product… *what?*

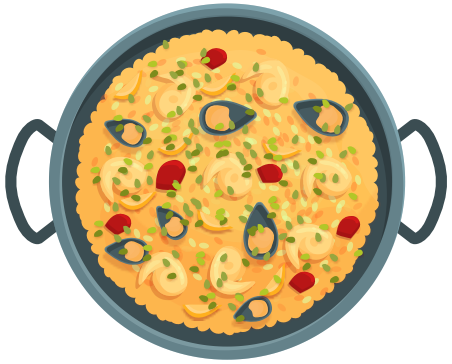
1. Variability and Software Product Lines

Paella!



1. Variability and Software Product Lines

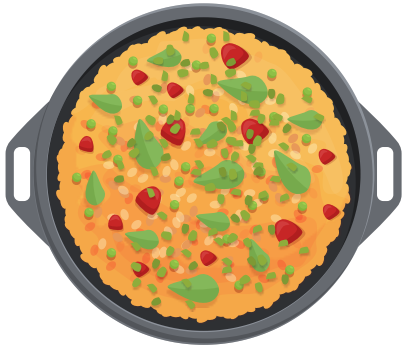
Paella!



Paella valenciana

Paella de marisco (seafood paella)

Paella de verduras (vegetable paella)



Arroz con cosas (rice with things)

Paella de arroz negro (seafood paella)

Paella de pollo (chicken paella)

1. Variability and Software Product Lines

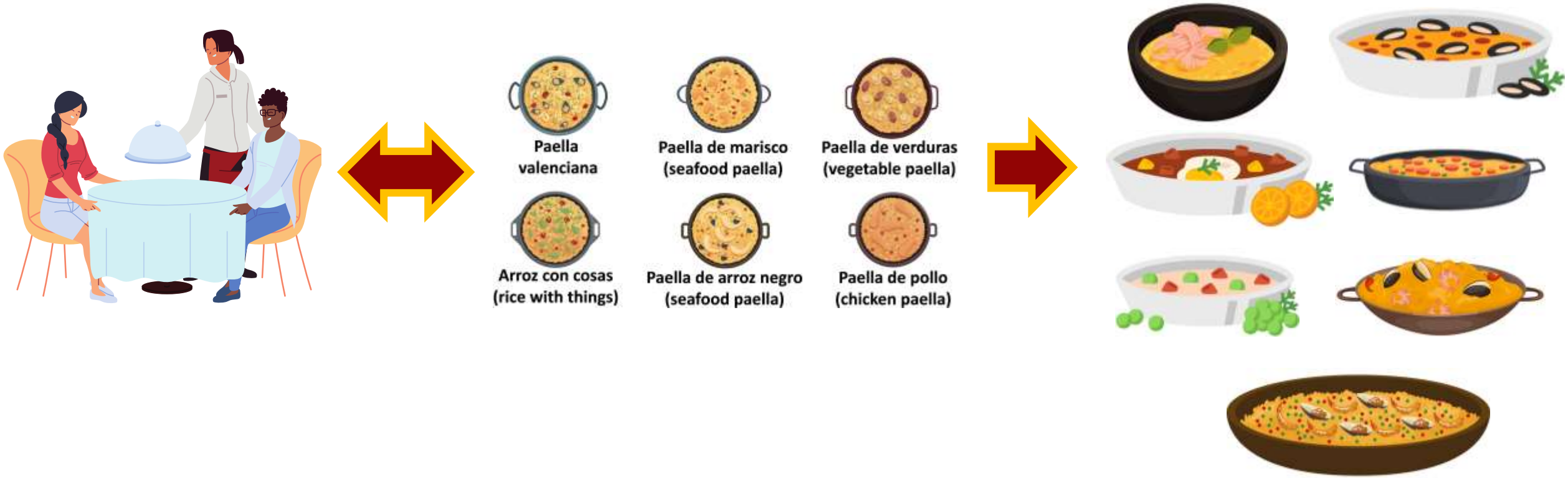
Mass production

Mass production

Producing efficiently a
large amount
of standarized products

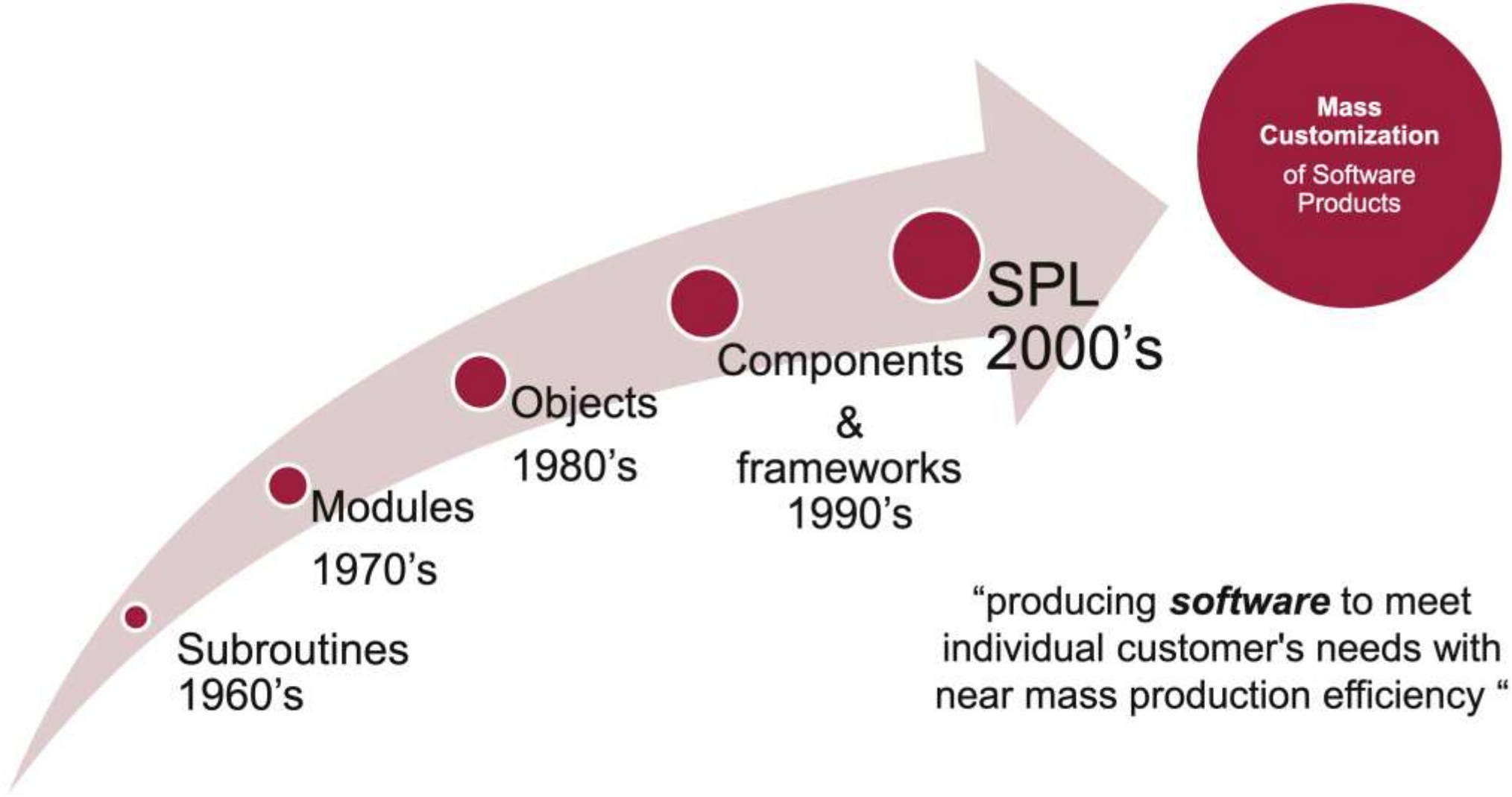
1. Variability and Software Product Lines

Mass customization – A set of product



1. Variability and Software Product Lines

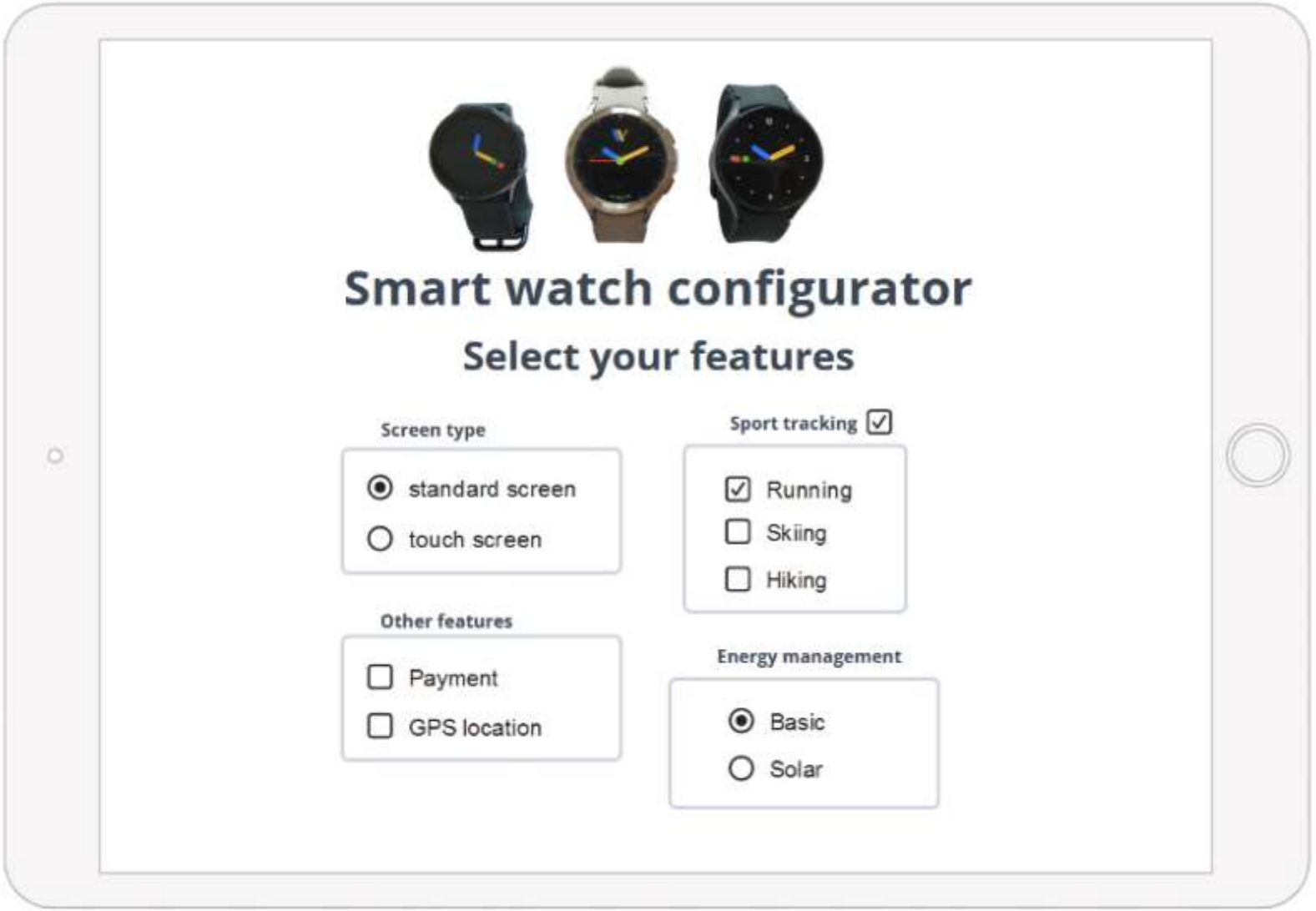
SPL



1. Variability and Software Product Lines
2. **Yet another language: UVL**
3. UVLHub and Open Science
4. flamapy
5. Conclusions

2. Yet another language: UVL

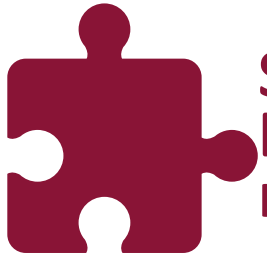
A typical example



2. Yet another language: UVL

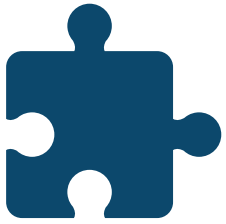
Variability Model

Common features

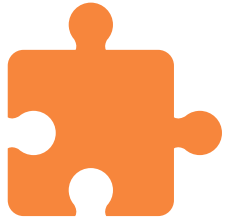


Screen
Energy
management

Variable features



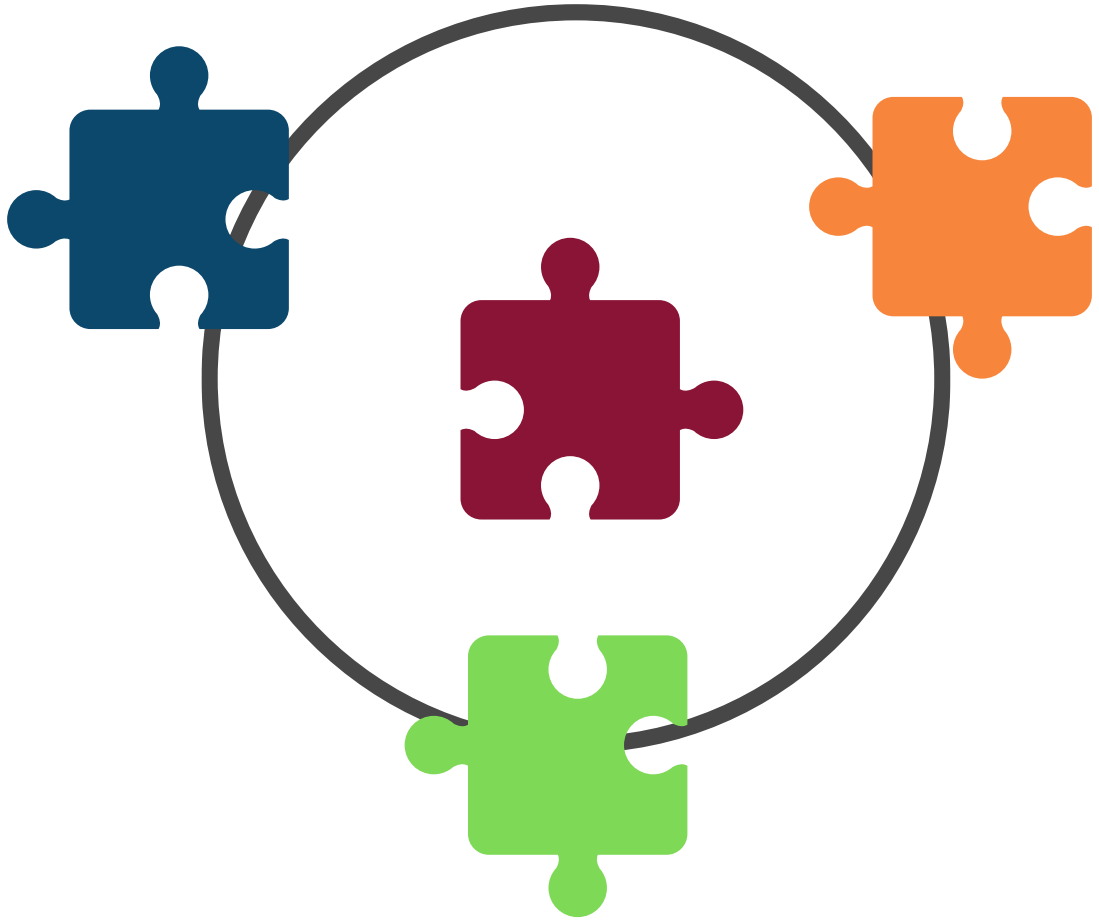
Payment



Sports
tracking

GPS

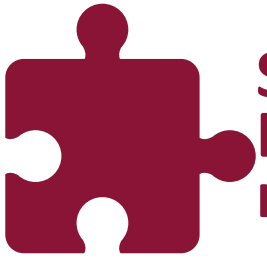
Variability Model



2. Yet another language: UVL

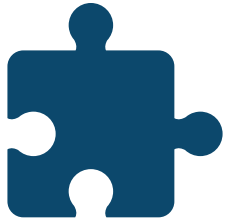
Variability Model

Common features



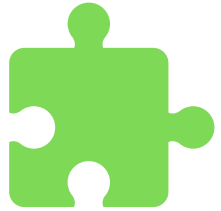
Screen
Energy management

Variable features



Payment

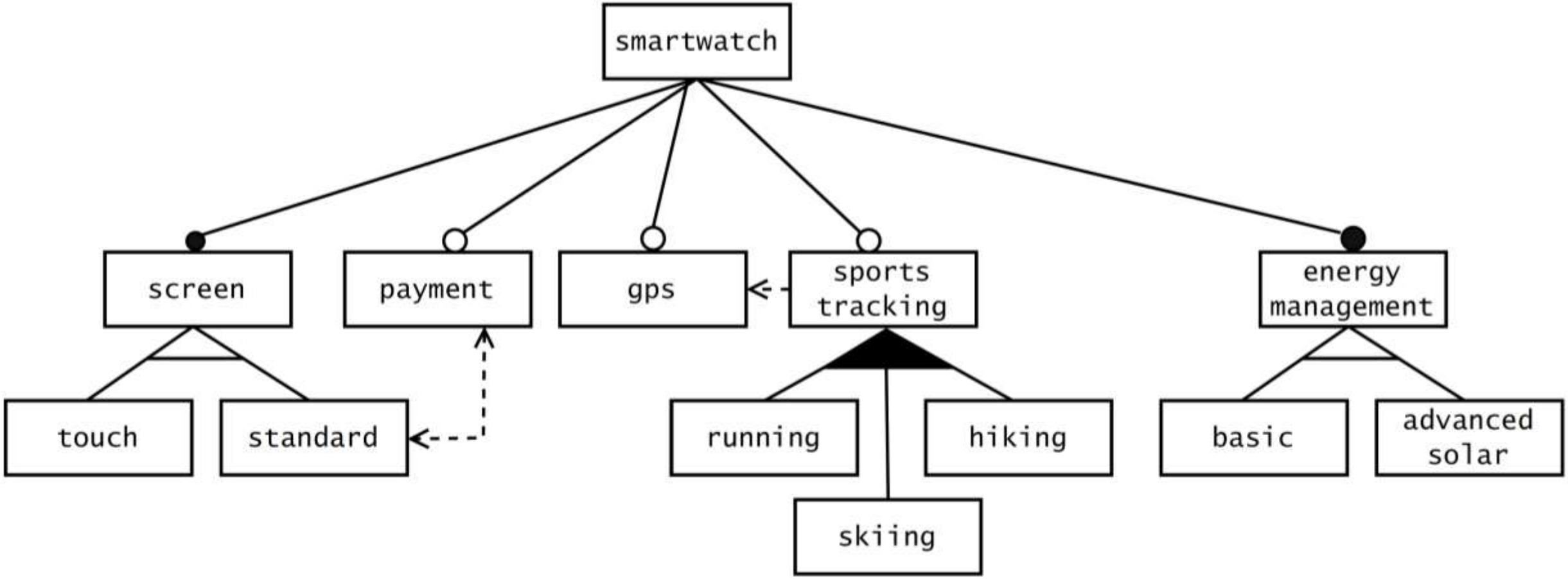
Sports tracking



GPS

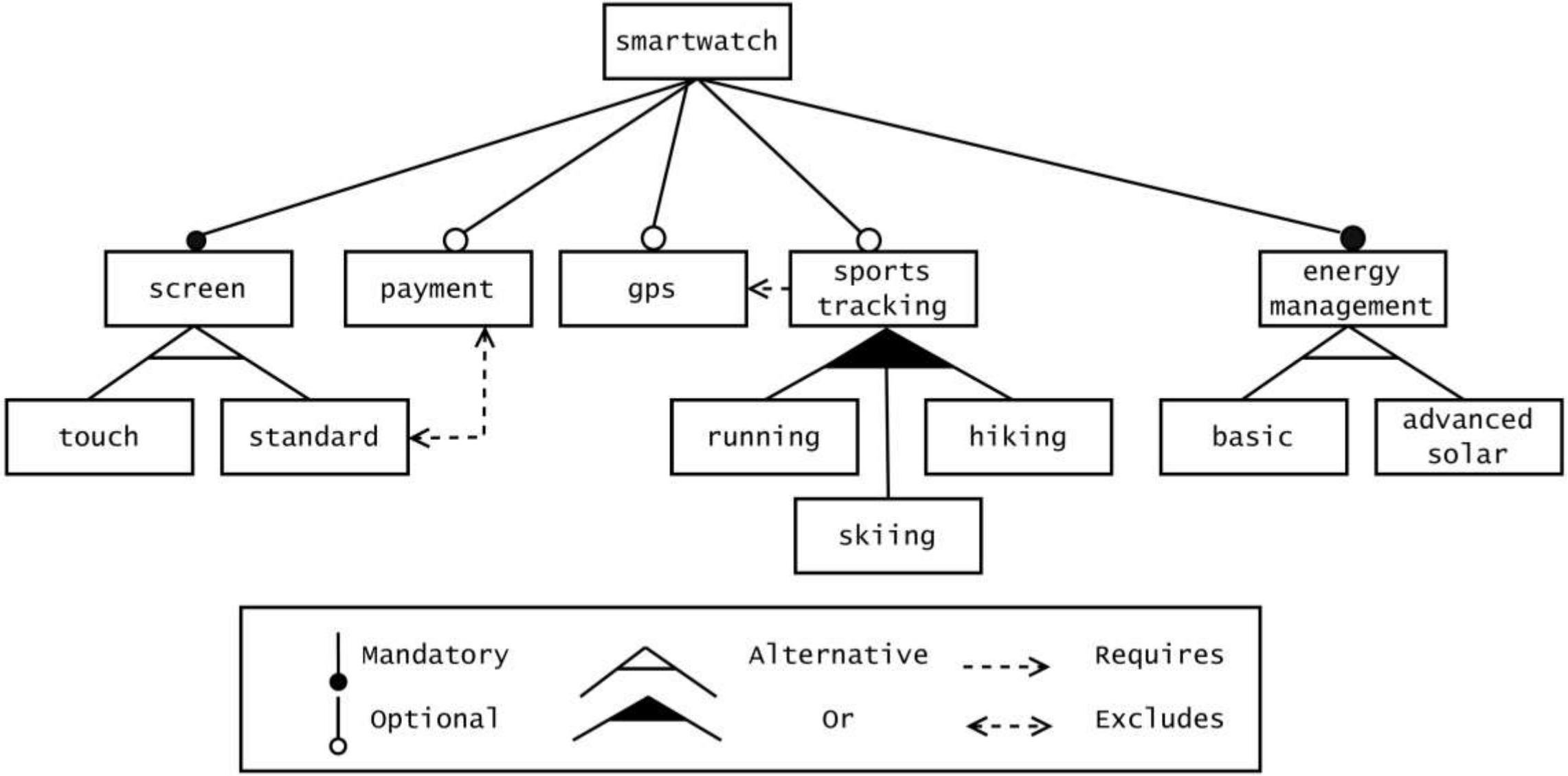
2. Yet another language: UVL

A typical example of feature model



2. Yet another language: UVL

A typical example of feature model



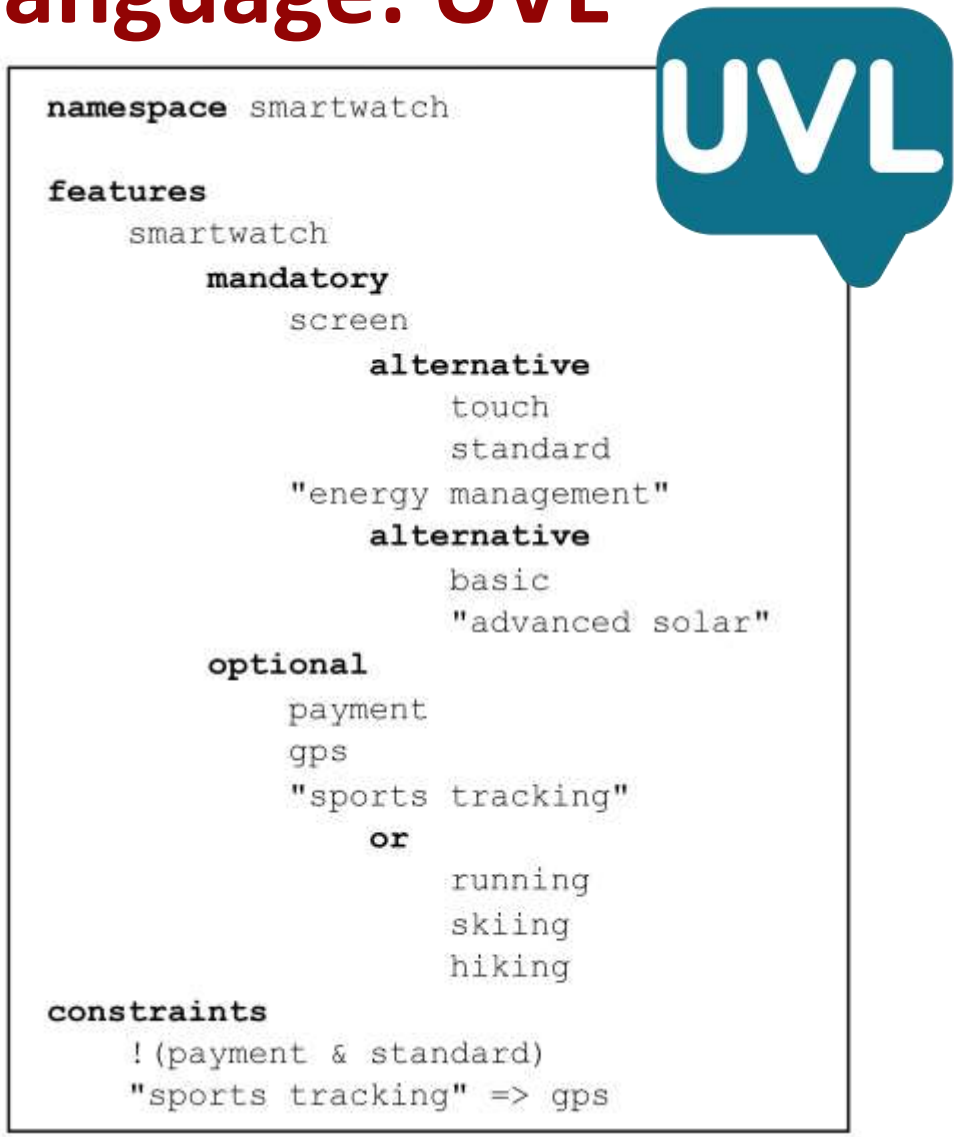
2. Yet another language: UVL

MODEVAR initiative

MODEVAR

International Workshops on Languages for Modelling Variability

2. Yet another language: UVL



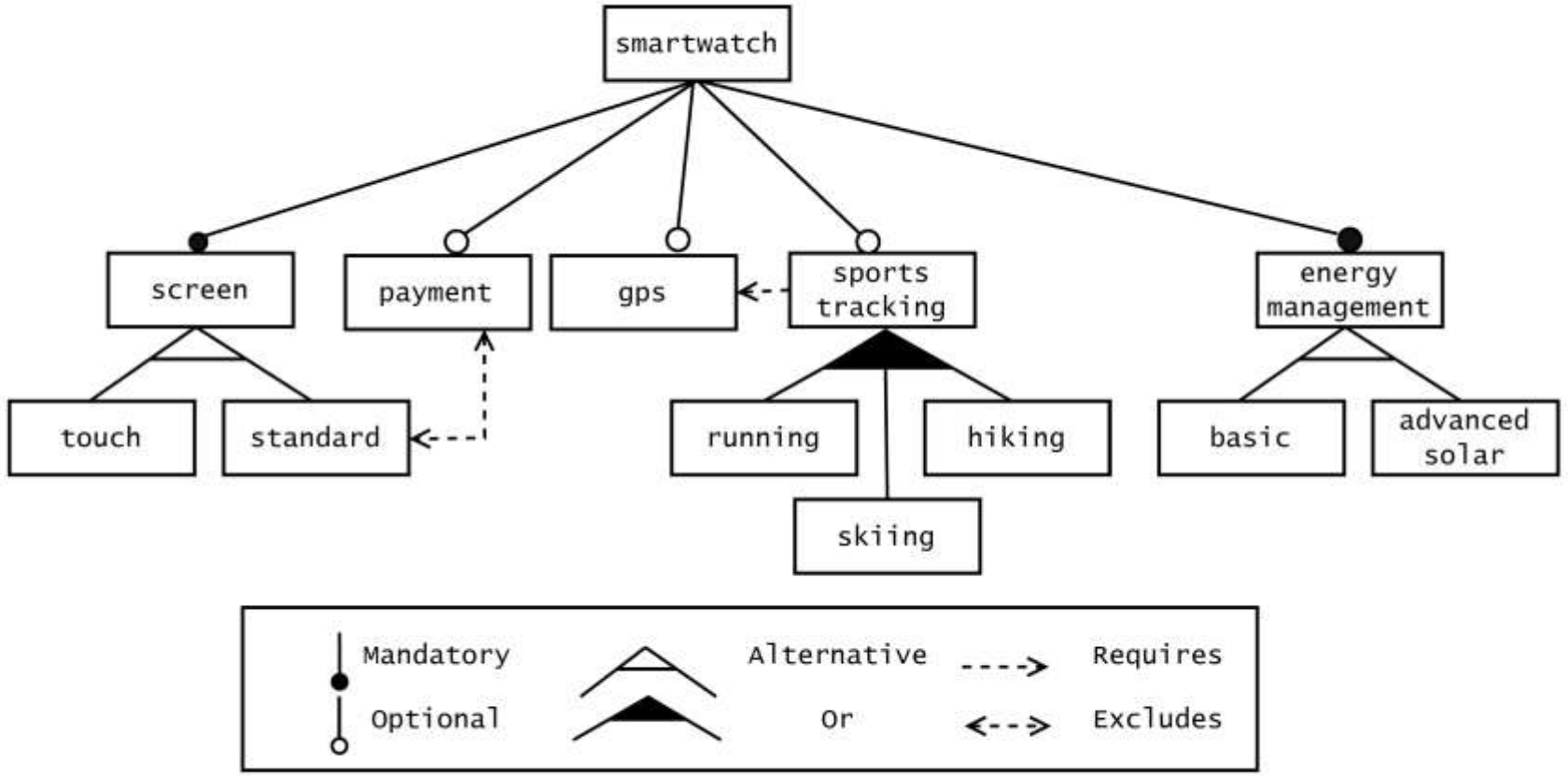
```
namespace smartwatch

features
    smartwatch
        mandatory
            screen
                alternative
                    touch
                    standard
            "energy management"
                alternative
                    basic
                    "advanced solar"
        optional
            payment
            gps
            "sports tracking"
                or
                    running
                    skiing
                    hiking
constraints
    !(payment & standard)
    "sports tracking" => gps
```

Universal Variability Language

2. Yet another language: UVL

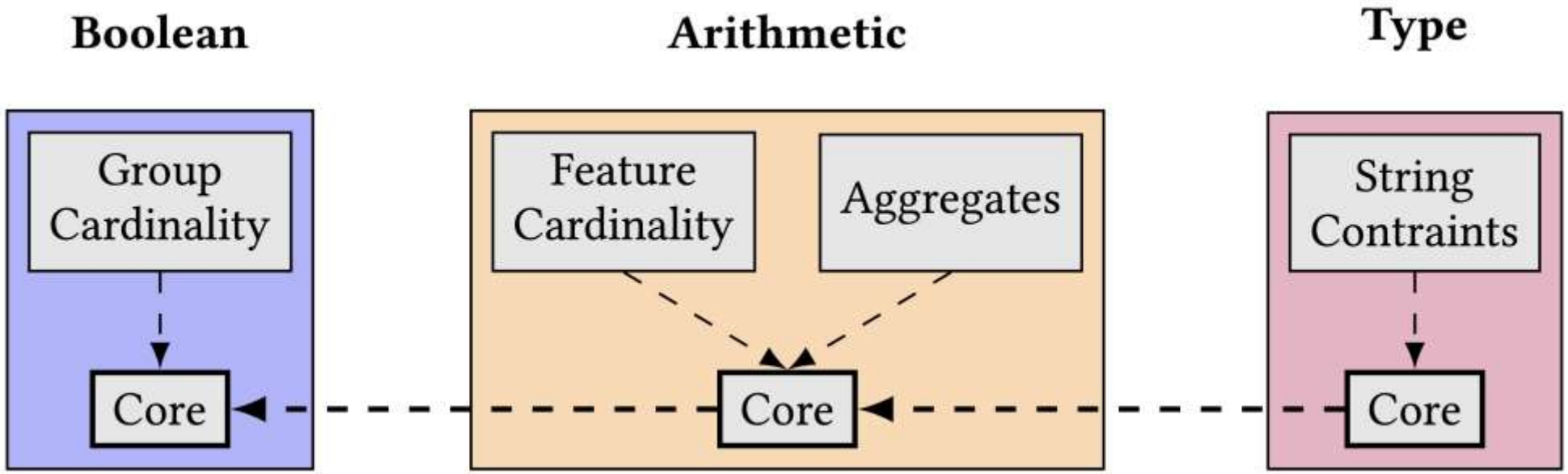
A typical example of UVL (smartwatch)

```
namespace smartwatch

features
    smartwatch
        mandatory
            screen
                alternative
                    touch
                    standard
            "energy management"
                alternative
                    basic
                    "advanced solar"
        optional
            payment
            gps
            "sports tracking"
                or
                    running
                    skiing
                    hiking
constraints
    !(payment & standard)
    "sports tracking" => gps
```



2. Yet another language: UVL

Language Level



Benavides, David and Sundermann, Chico and Feichtinger, Kevin and Galindo, José A. and Rabiser, Rick and Thüm, Thomas, Uvl: Feature Modelling with the Universal Variability Language. Available at SSRN: https://ssrn.com/abstract=4764657 or http://dx.doi.org/10.2139/ssrn.4764657

Chico Sundermann, Stefan Vill, Thomas Thüm, Kevin Feichtinger, Prankur Agarwal, Rick Rabiser, José A. Galindo, and David Benavides. 2023. UVL- Parser: Extending UVL with Language Levels and Conversion Strategies. In 27th ACM International Systems and Software Product Line Conference - Volume B (SPLC '23), August 28-September 1, 2023, Tokyo, Japan. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3579028.3609013

2. Yet another language: UVL

An example of arithmetic level


```
features
    smartwatch
    mandatory
        screen
            alternative
                touch {Power 3}
                standard {Power 2}
        "energy management" {Power 5}
optional
        payment {Power 4}
        gps {Power 2}
        Integer Watt
        "sleep tracking" {Power 2}
constraints
    sum(Power) <= Watt
```

1. Variability and Software Product Lines
2. Yet another language: UVL
3. **UVLHub and Open Science**
4. flamapy
5. Conclusions

3. UVLHub and Open Science

UVL dataset

3. UVLHub and Open Science

Dataset in science

Lots of datasets!

They are IMPORTANT because...

- Evidence base
- Reproducibility
- New findings
- Collaboration
- Transparency
- Education

3. UVLHub and Open Science

Dataset in science

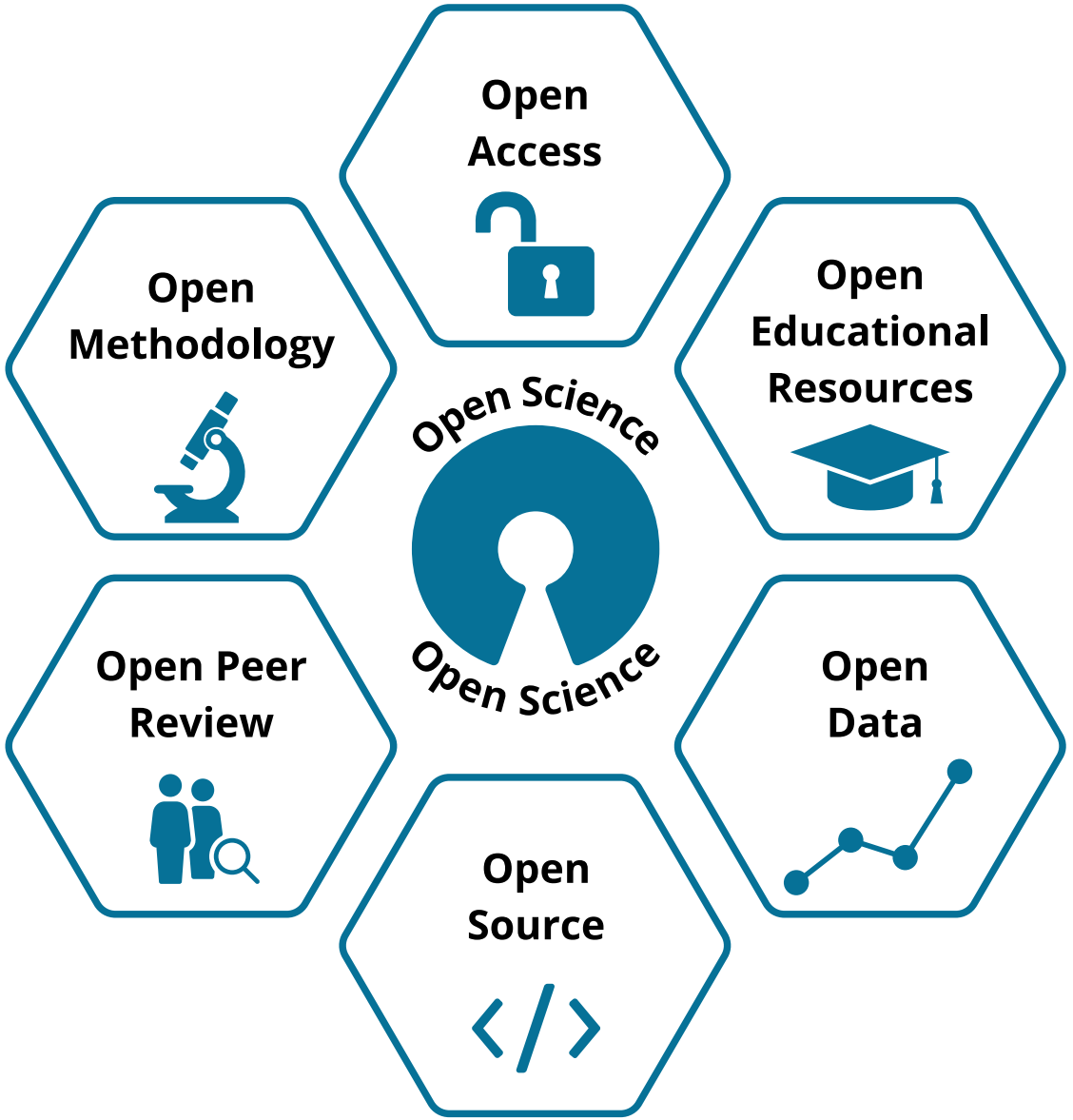
Private repositories

Lost, corrupt, incomplete...

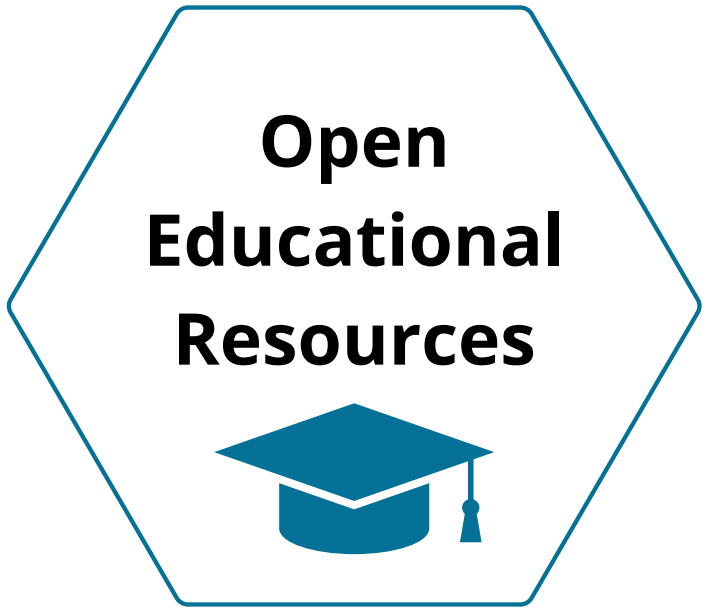
System down

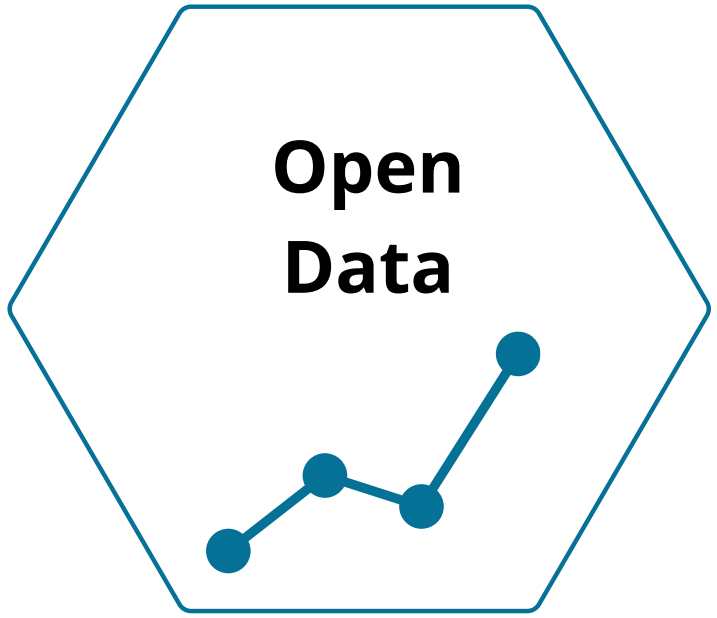
3. UVLHub and Open Science

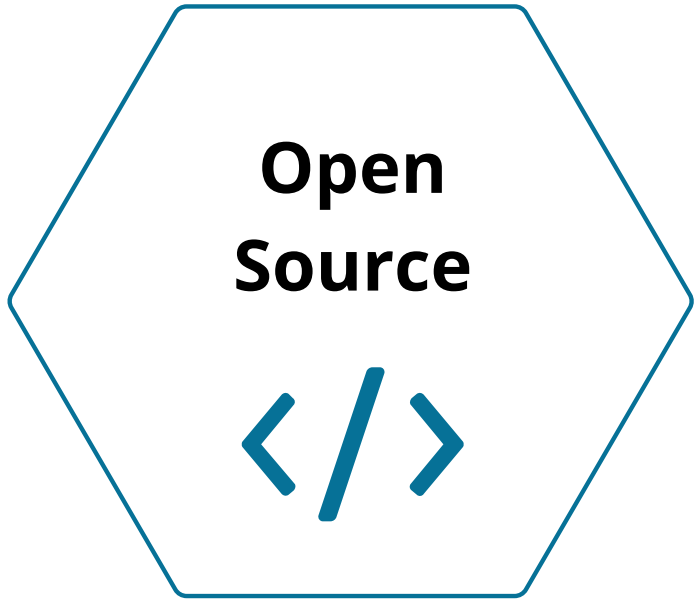
Key Principles



3. UVLHub and Open Science







3. UVLHub and Open Science

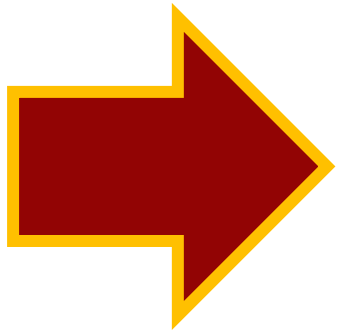
Can The Open Science provide a solution to my UVL dataset problem?

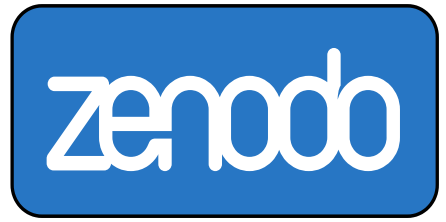
3. UVLHub and Open Science

Different initiatives

Open Access Infrastructure for Research in Europe

Research Data Management (CERN)

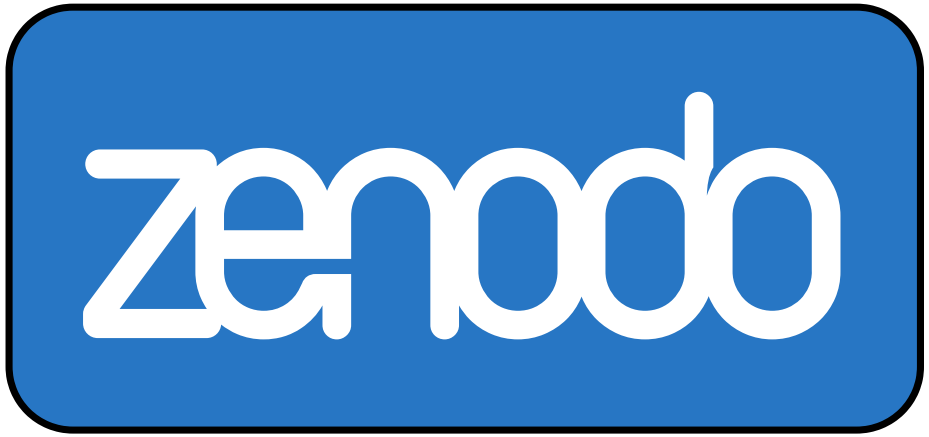




Open Access Repository

3. UVLHub and Open Science

Different initiatives



But in Zenodo you can load **ANY** kind of dataset, not only UVL...

3. UVLHub and Open Science

How do I control that only **UVL datasets** are uploaded?

3. UVLHub and Open Science

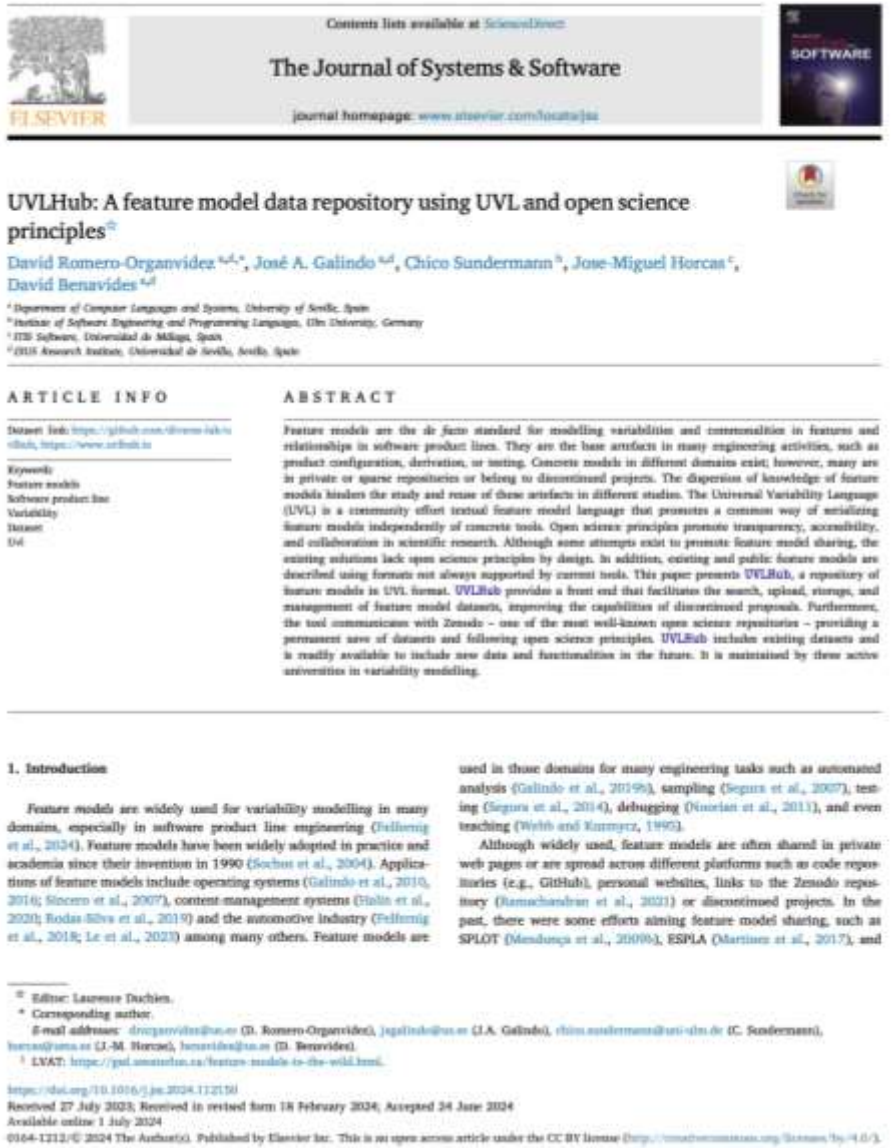
uvlhub.io

A feature model repository using UVL and Open Science principles

UNIVERSIDAD D SEVILLA 1505

UNIVERSIDAD DE MÁLAGA

universität ulm



Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

UVLHub: A feature model data repository using UVL and open science principles☆

David Romero-Organvidez [a,d,*], José A. Galindo [a,d], Chico Sundermann [b], Jose-Miguel Horcas [c], David Benavides [a,d]

[a] Department of Computer Languages and Systems, University of Seville, Spain
[b] Institute of Software Engineering and Programming Languages, Ulm University, Germany
[c] ITIS Software, Universidad de Málaga, Spain
[d] SCORE Research Institute, Universidad de Sevilla, Seville, Spain

ARTICLE INFO

Dataset link: https://github.com/diverso-lab/uvlhub, https://www.uvlhub.io

Keywords:
Feature models
Software product line
Variability
Dataset
UVL

ABSTRACT

Feature models are the de facto standard for modelling variabilities and commonalities in features and relationships in software product lines. They are the base artifacts in many engineering activities, such as product configuration, derivation, or testing. Concrete models in different domains exist; however, many are in private or sparse repositories or belong to discontinued projects. The dispersion of knowledge of feature models hinders the study and reuse of these artefacts in different studies. The Universal Variability Language (UVL) is a community effort textual feature model language that promotes a common way of serialising feature models independently of concrete tools. Open science principles promote transparency, accessibility, and collaboration in scientific research. Although some attempts exist to promote feature model sharing, the existing solutions lack open science principles by design. In addition, existing and public feature models are described using formats not always supported by current tools. This paper presents UVLHub, a repository of feature models in UVL format. UVLHub provides a front end that facilitates the search, upload, storage, and management of feature model datasets, improving the capabilities of discontinued proposals. Furthermore, the tool communicates with Zenodo – one of the most well-known open science repositories – providing a permanent save of datasets and following open science principles. UVLHub includes existing datasets and is readily available to include new data and functionalities in the future. It is maintained by three active universities in variability modelling.

1. Introduction

Feature models are widely used for variability modelling in many domains, especially in software product line engineering (Fellmeng et al., 2024). Feature models have been widely adopted in practice and academia since their invention in 1990 (Sochos et al., 2004). Applications of feature models include operating systems (Galindo et al., 2016; Sincero et al., 2007), content-management systems (Halin et al., 2020; Rodas-Silva et al., 2019) and the automotive industry (Fellmeng et al., 2018; Le et al., 2023) among many others. Feature models are used in those domains for many engineering tasks such as automated analysis (Galindo et al., 2019b), sampling (Segura et al., 2007), testing (Segura et al., 2014), debugging (Nicolas et al., 2011), and even teaching (Webb and Kuzniarz, 1995).

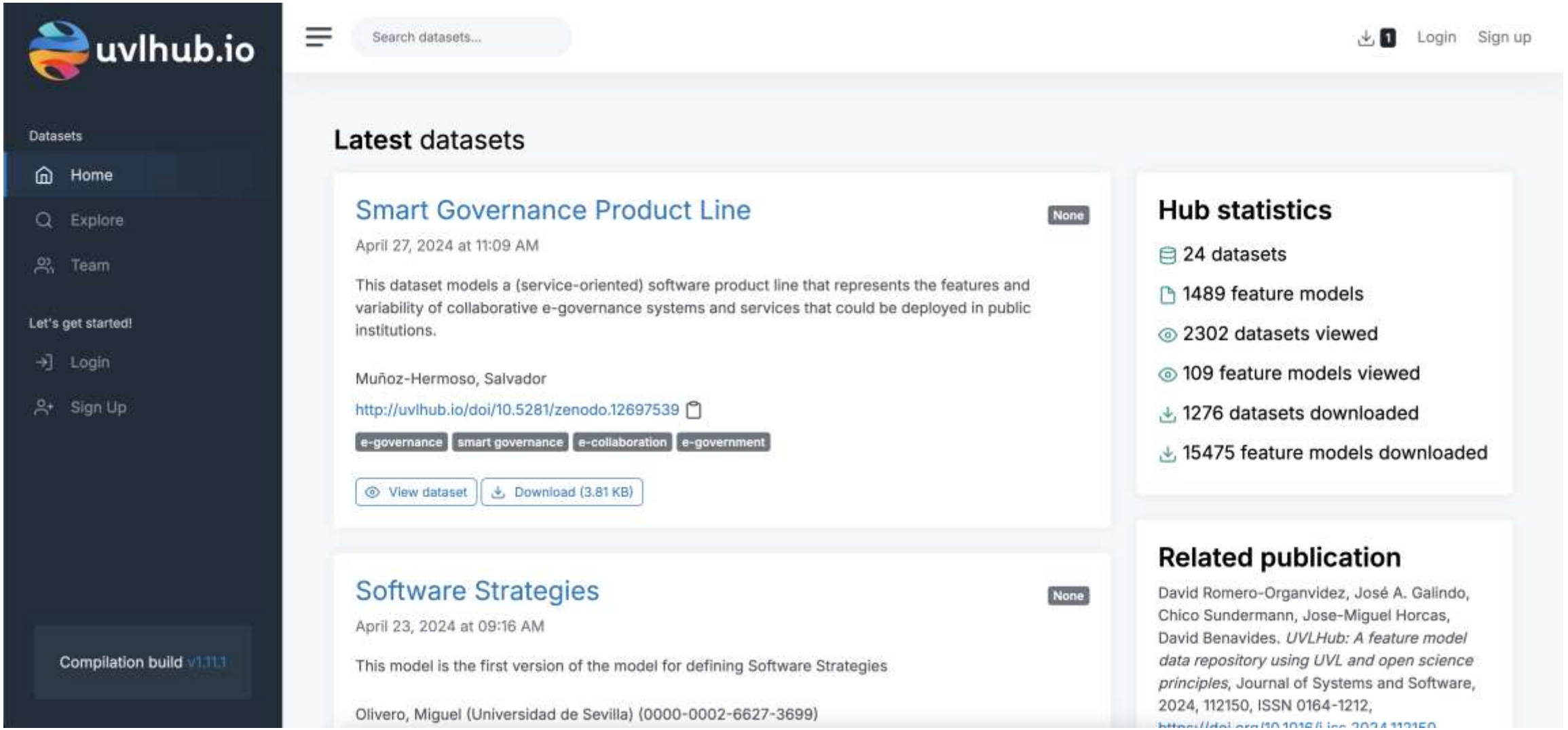
Although widely used, feature models are often shared in private web pages or are spread across different platforms such as code repositories (e.g., GitHub), personal websites, links to the Zenodo repository (Ramachandran et al., 2021) or discontinued projects. In the past, there were some efforts aiming feature model sharing, such as SPLOT (Mendonça et al., 2009b), ESPLA (Martínez et al., 2017), and

☆ Editor: Laurence Duchien.
* Corresponding author.
E-mail addresses: drorganvidez@us.es (D. Romero-Organvidez), jagalindo@us.es (J.A. Galindo), chico.sundermann@uni-ulm.de (C. Sundermann), horcas@uma.es (J.-M. Horcas), benavides@us.es (D. Benavides).
[1] LVAT: https://gsd.uwaterloo.ca/feature-models-in-the-wild.html.

https://doi.org/10.1016/j.jss.2024.112150
Received 27 July 2023; Received in revised form 18 February 2024; Accepted 24 June 2024
Available online 1 July 2024

3. UVLHub and Open Science

Sneak peek



3. UVLHub and Open Science

3. UVLHub and Open Science

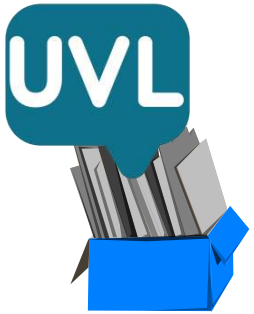
zenodo

uvlhub.io

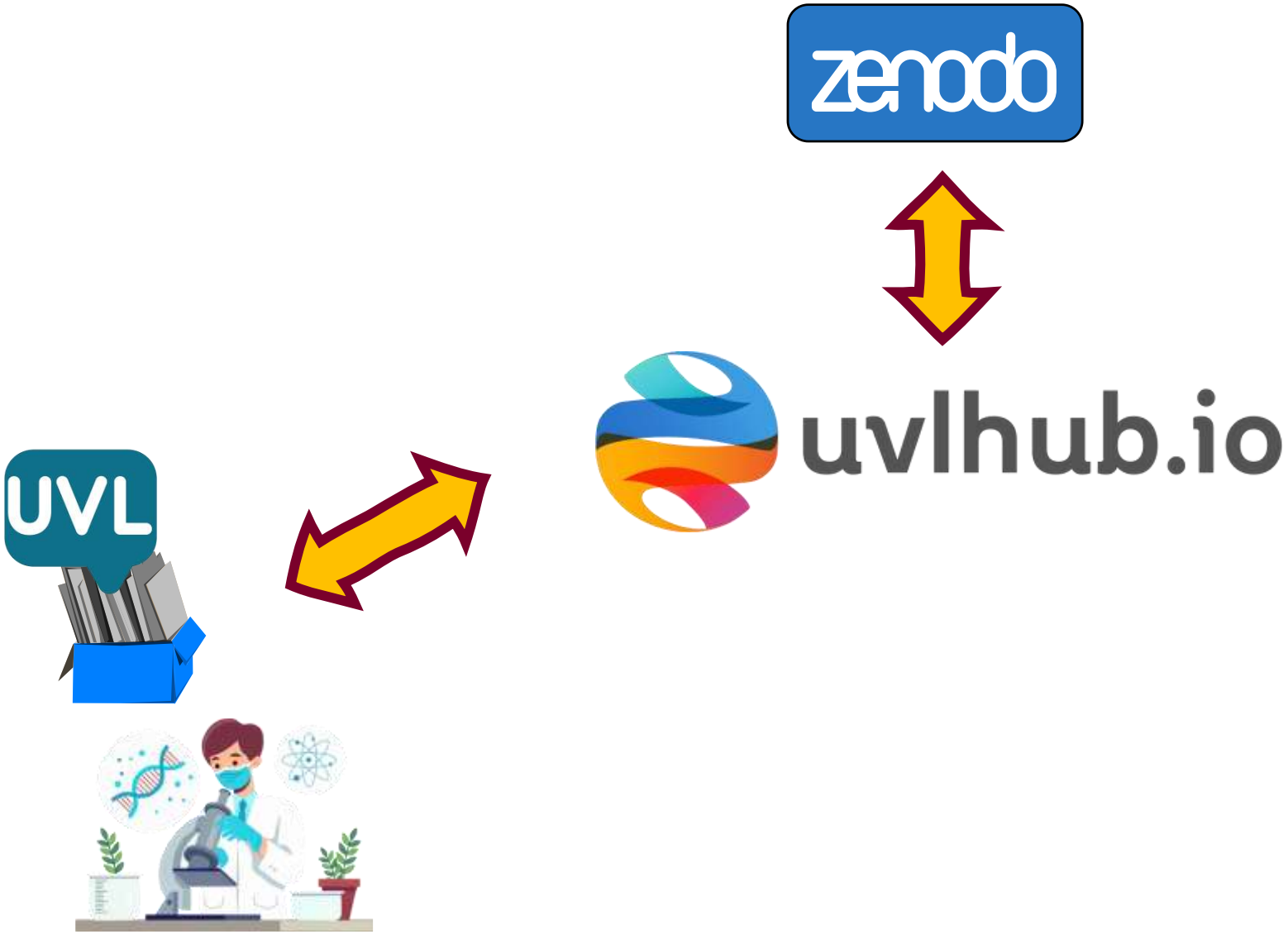
3. UVLHub and Open Science

zenodo

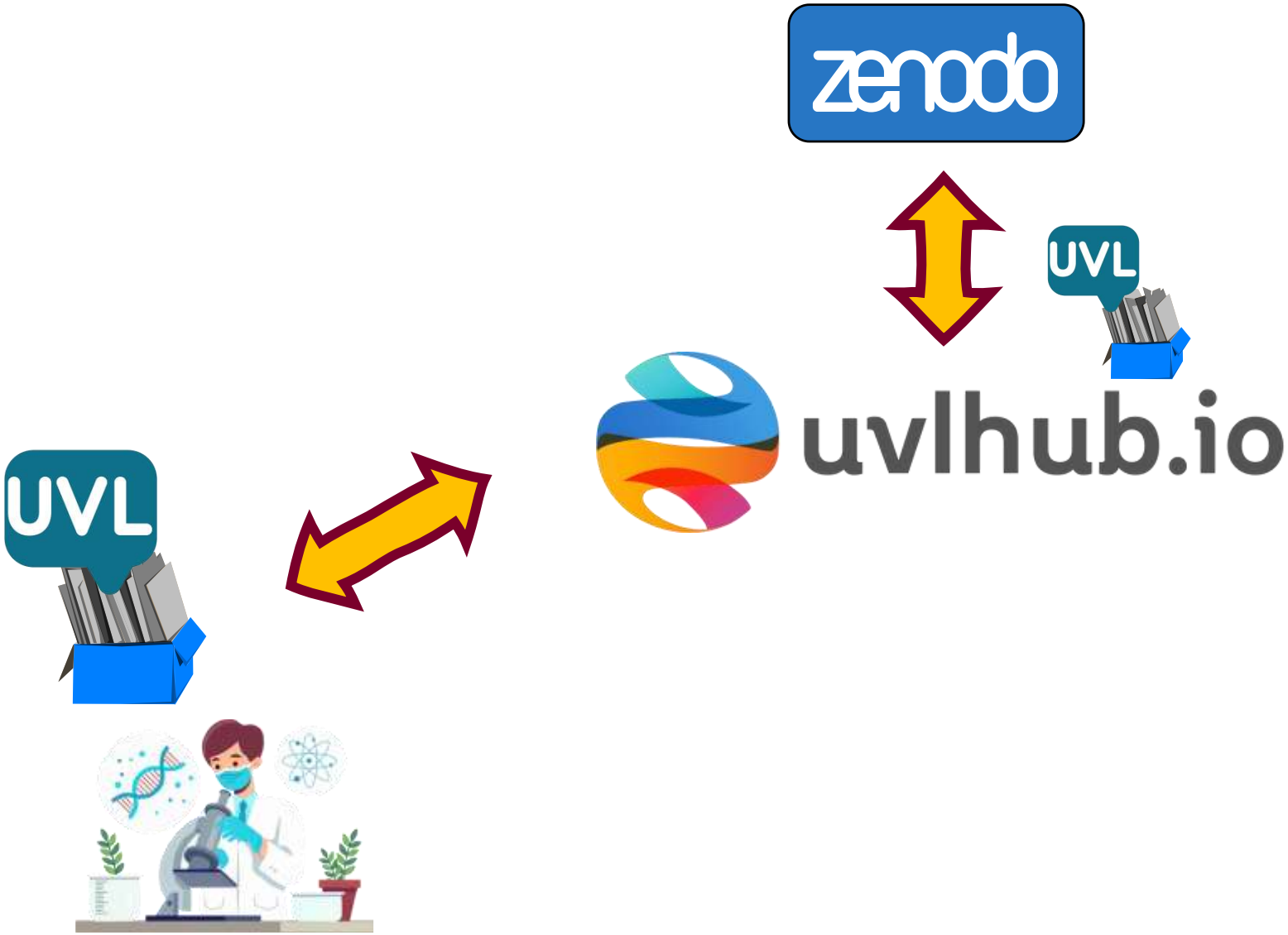
uvlhub.io



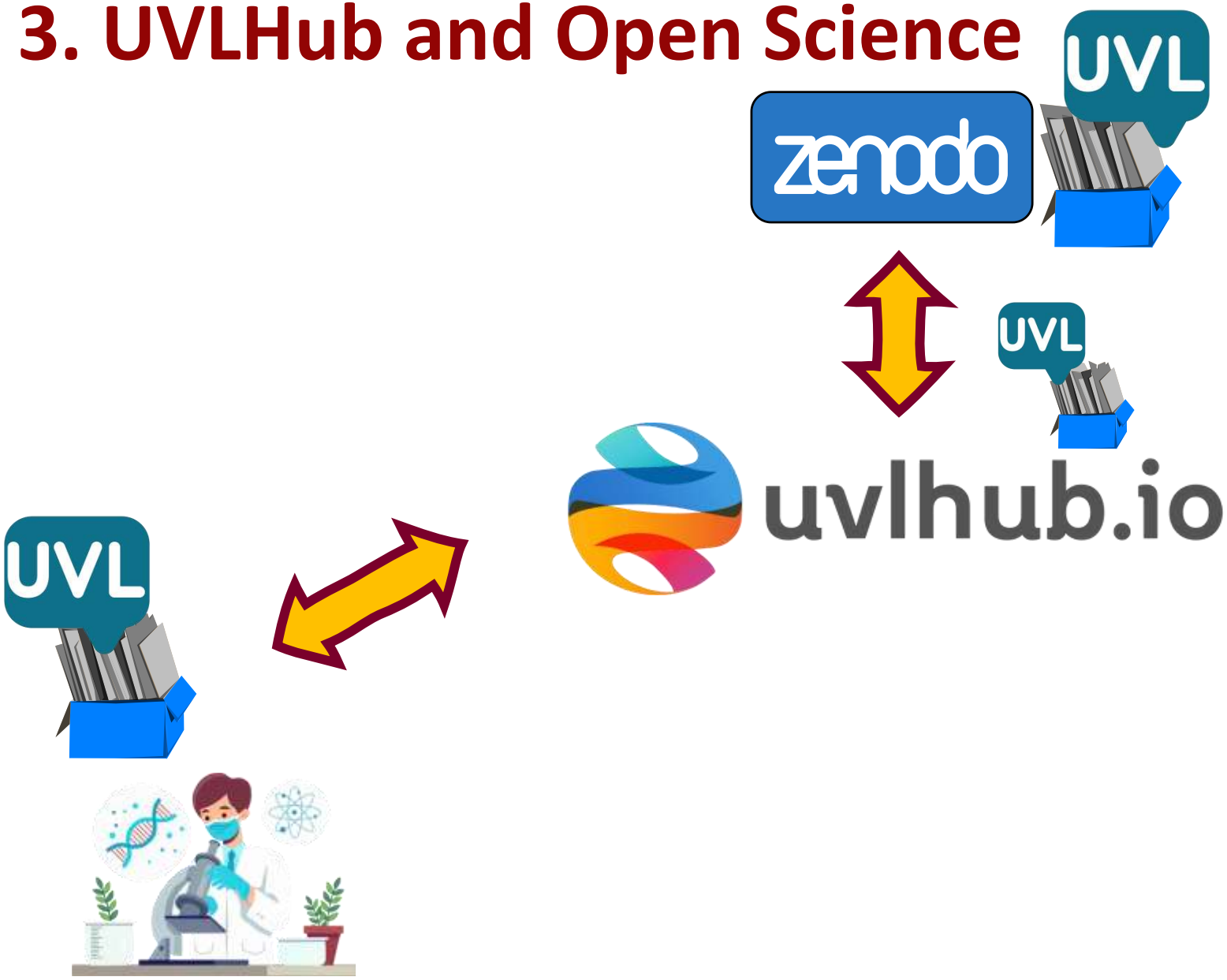
3. UVLHub and Open Science



3. UVLHub and Open Science

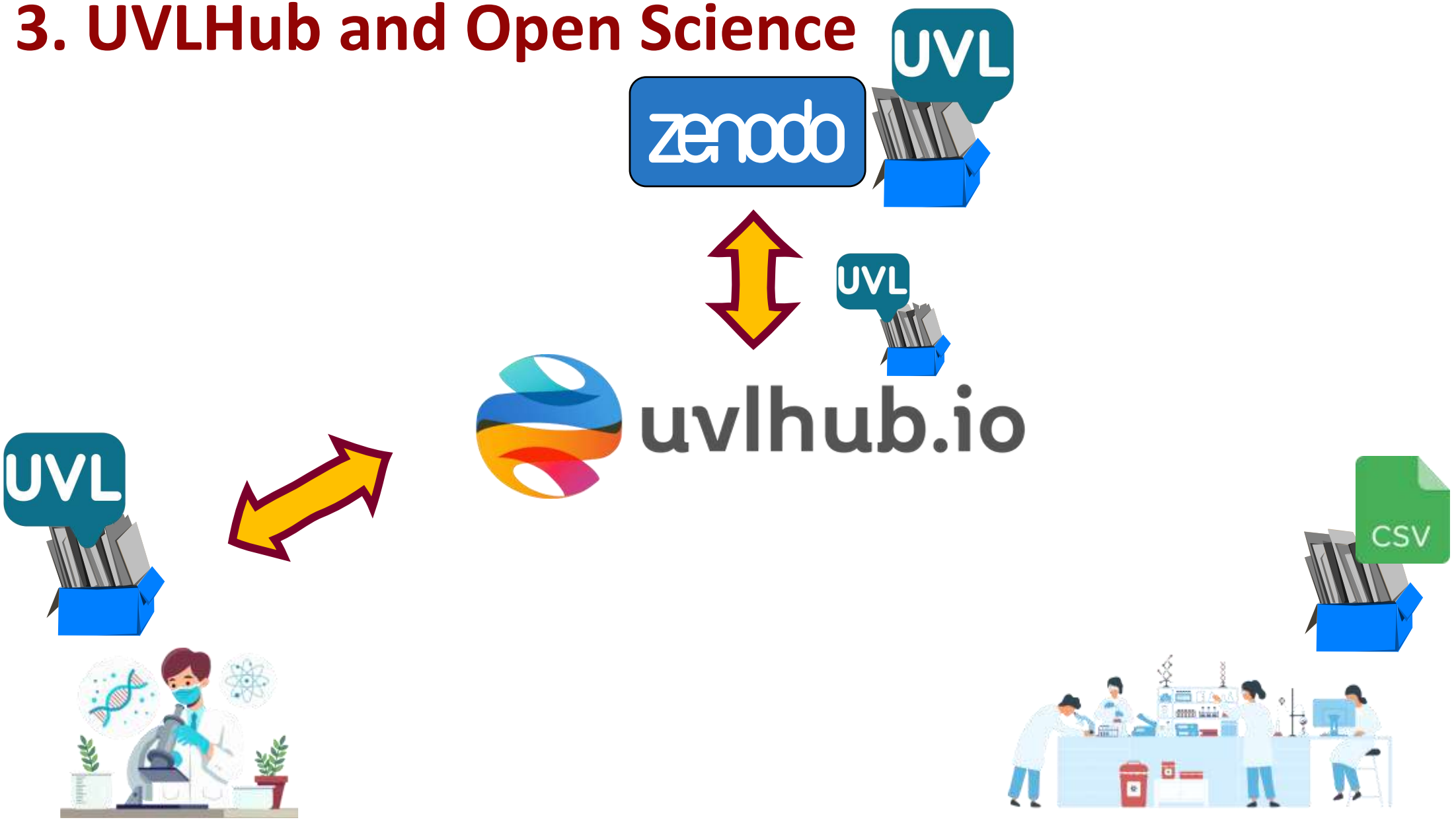


3. UVLHub and Open Science

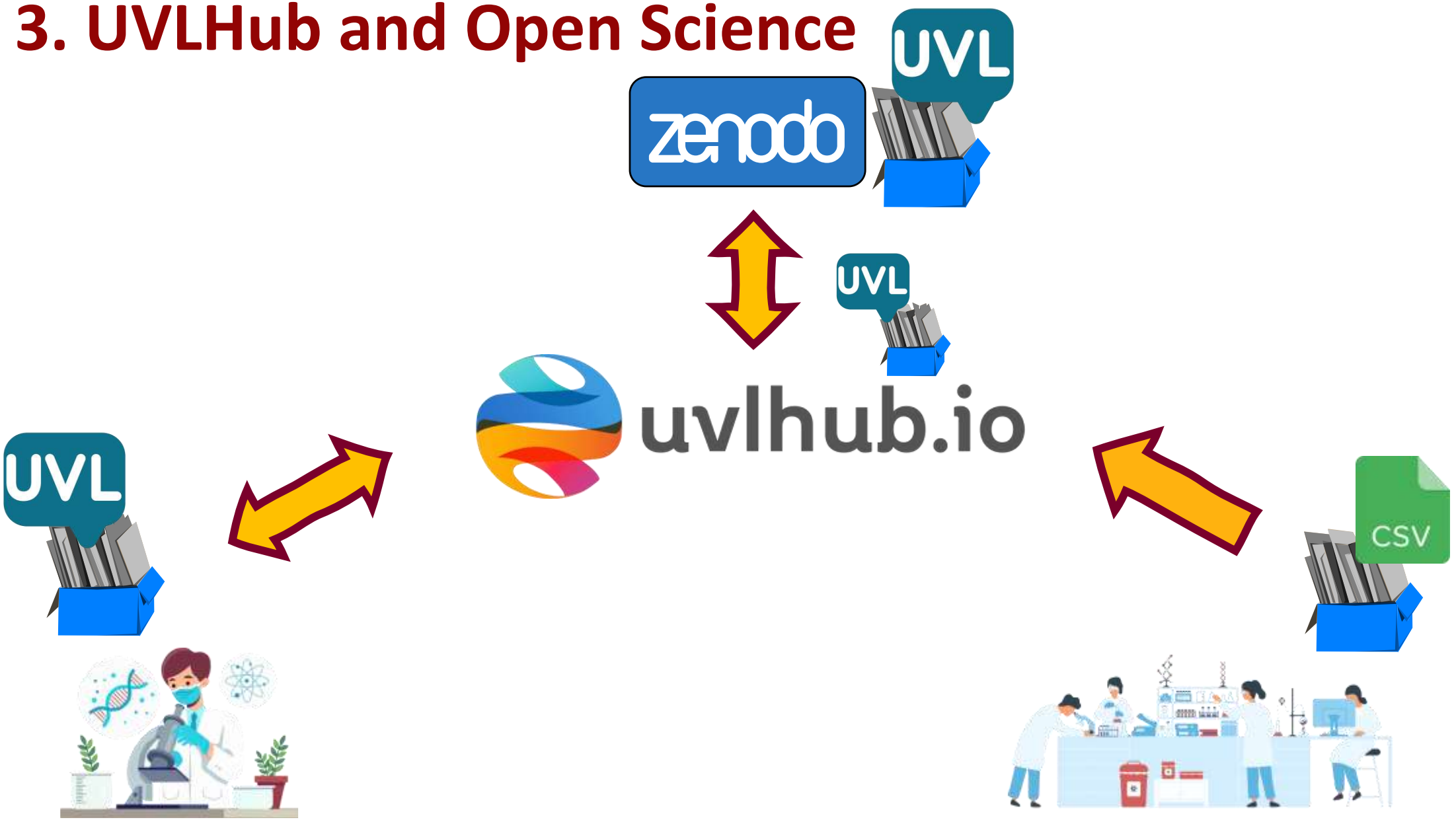


3. UVLHub and Open Science

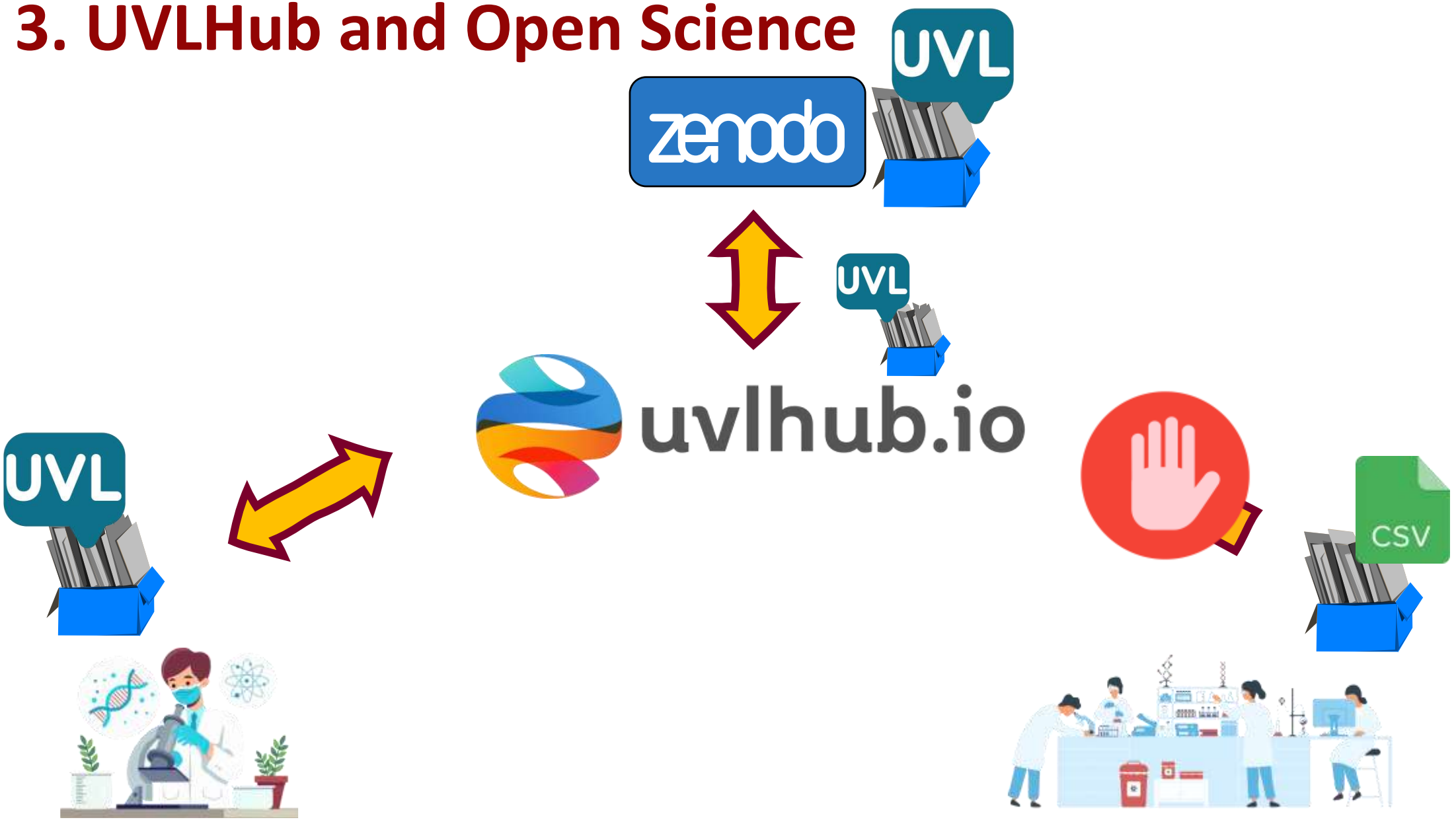
3. UVLHub and Open Science



3. UVLHub and Open Science



3. UVLHub and Open Science

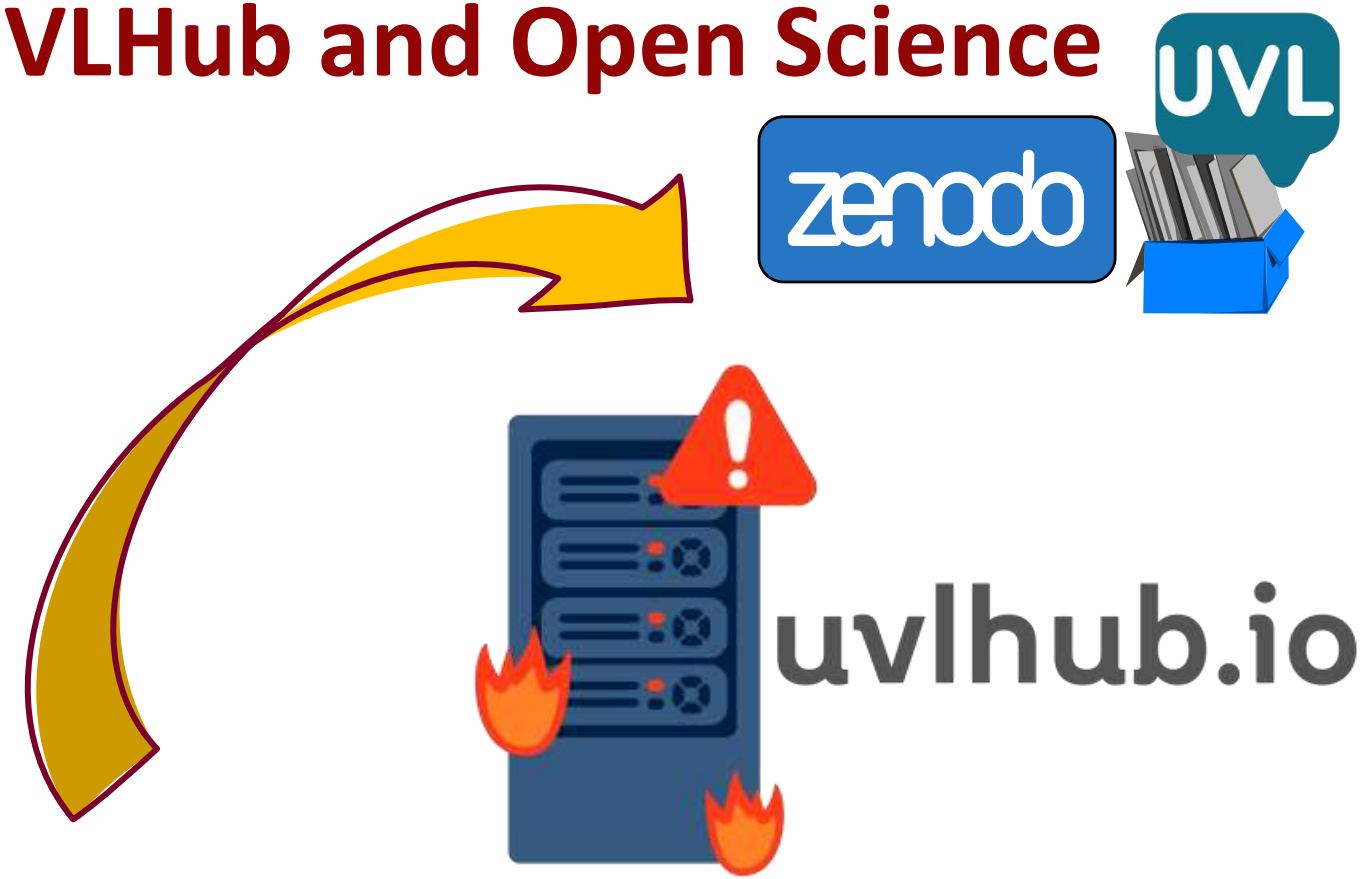


3. UVLHub and Open Science

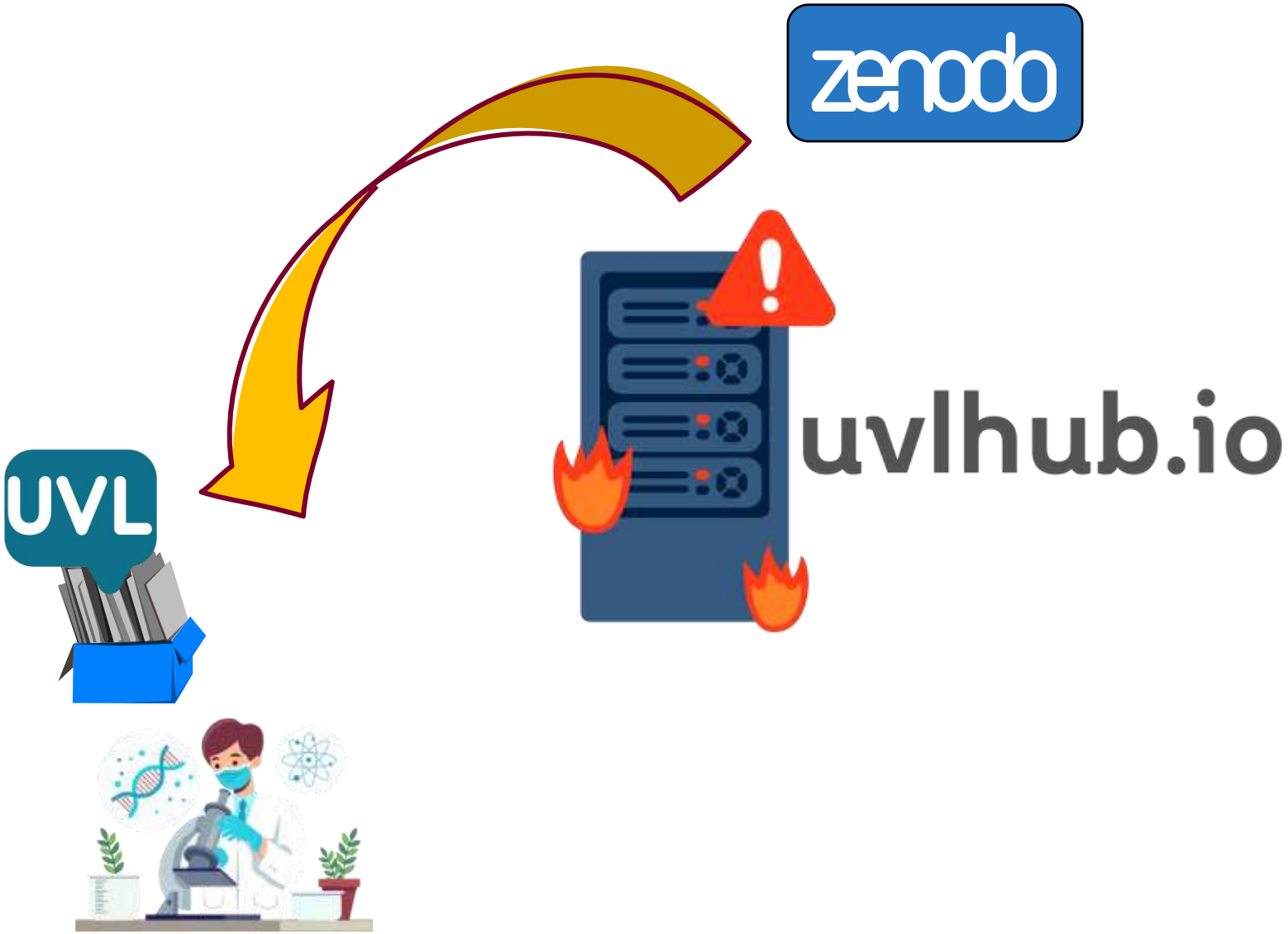
zenodo

uvlhub.io

3. UVLHub and Open Science

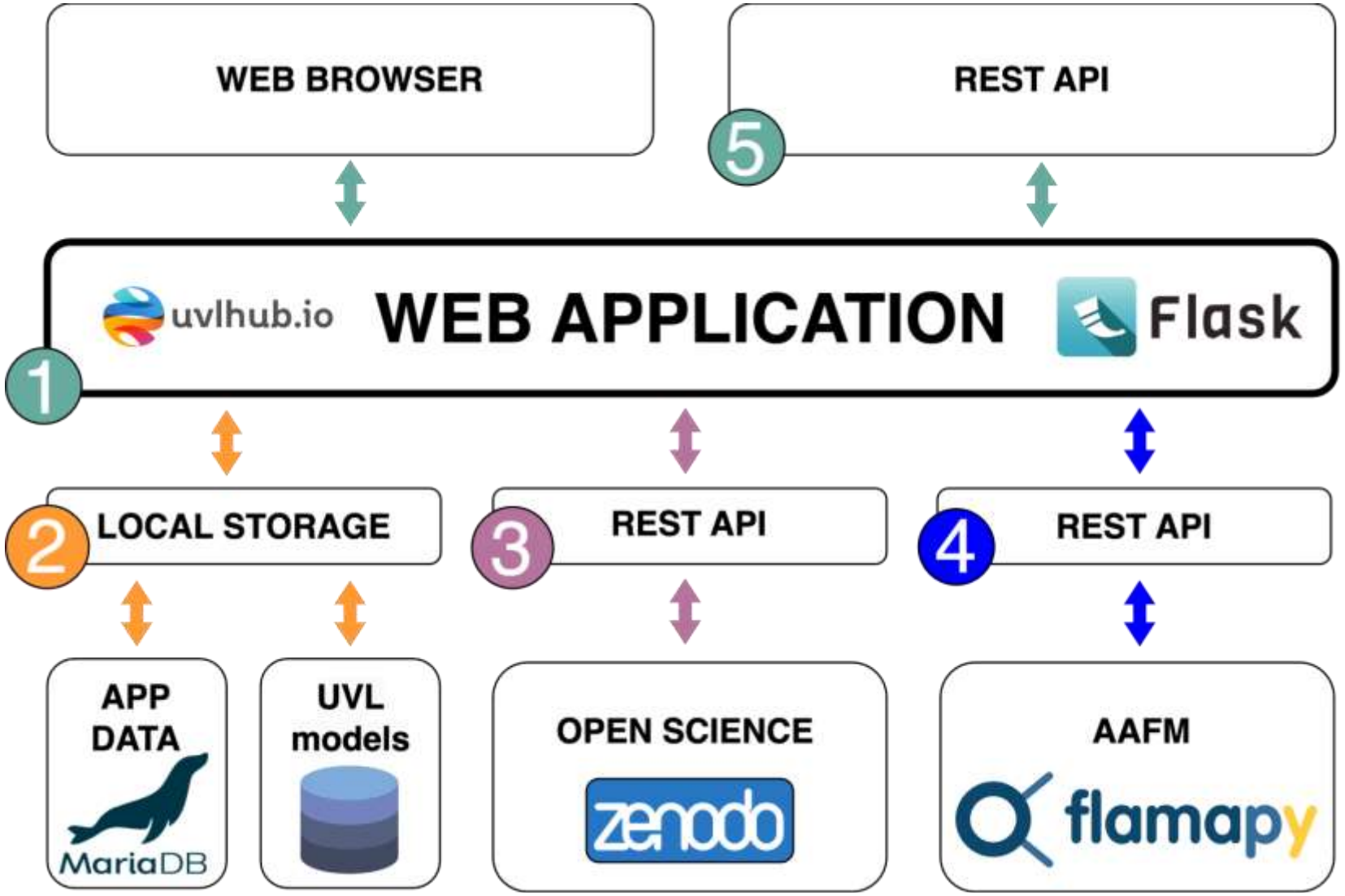


3. UVLHub and Open Science



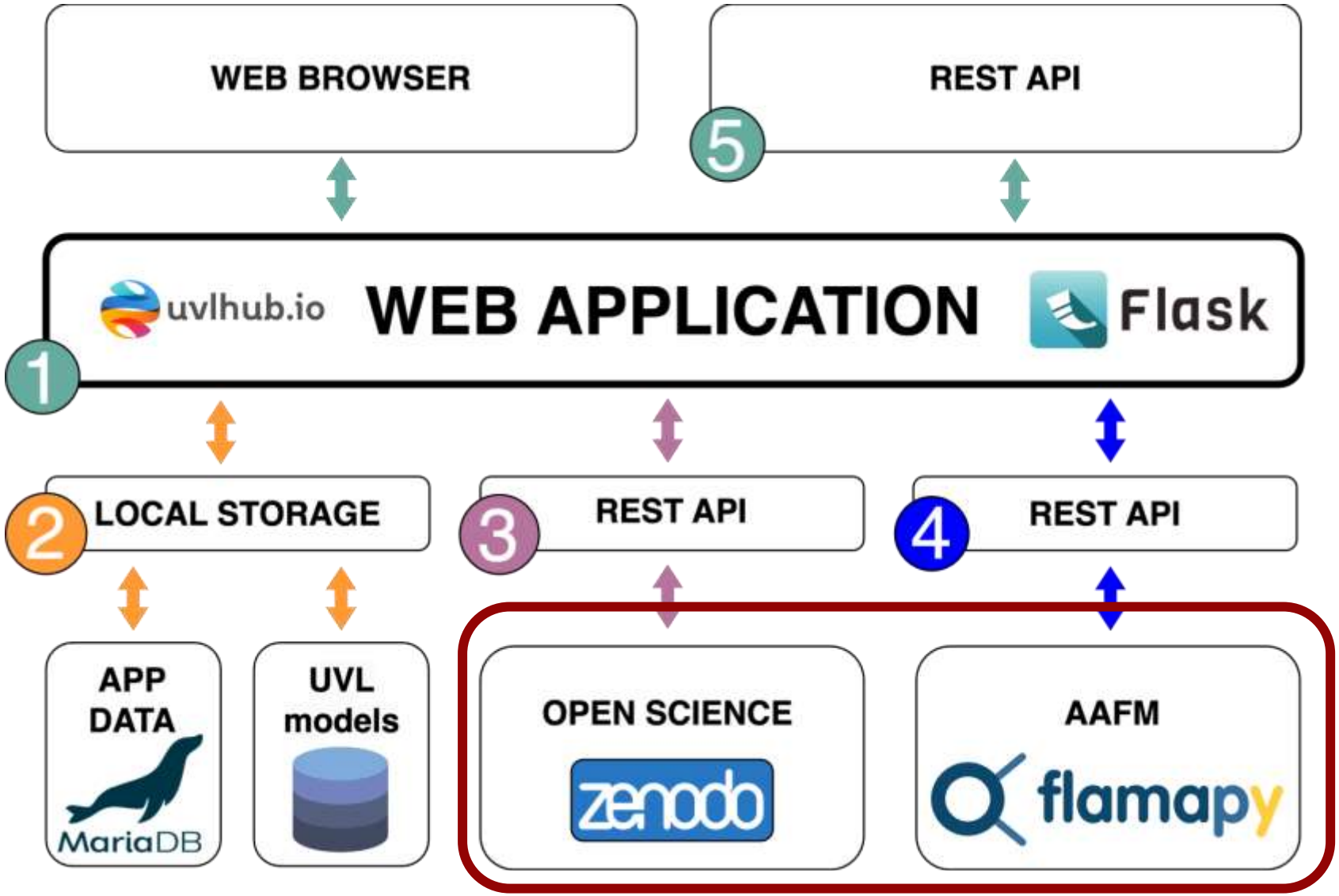
3. UVLHub and Open Science

Architecture



3. UVLHub and Open Science

Architecture



3. UVLHub and Open Science

I can adapt it to other contexts

Thanks to the uvlhub layer, I have...

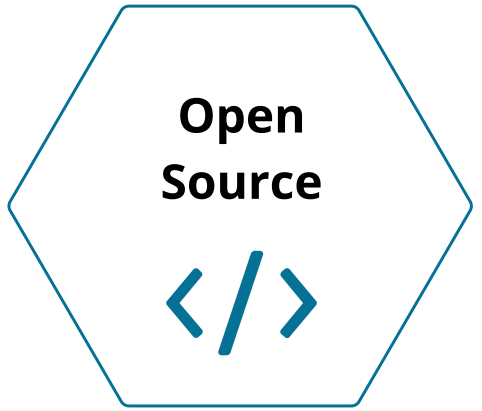
- **Filtering**
- **Analysis**
- **Search**
- **Business rules**
- **Domain-specific features**

1. Yet another language: UVL
2. Variability and Software Product Lines
3. UVLHub and Open Science
4. **flamapy**
5. Conclusions

4. flamapy

The cutting-edge Python-based tool for
Automated Analysis of Feature Models using UVL
and more

flamapy.org



4. flamapy

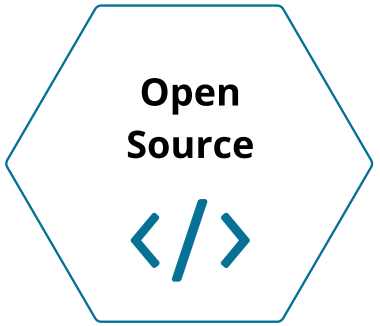
A tool and a framework

As a tool

- Allows to analyze feature models
- Multiple interfaces

As a framework

- Adapt it to what you need
- Integrate it in your tools



4. flamapy

Supported formats

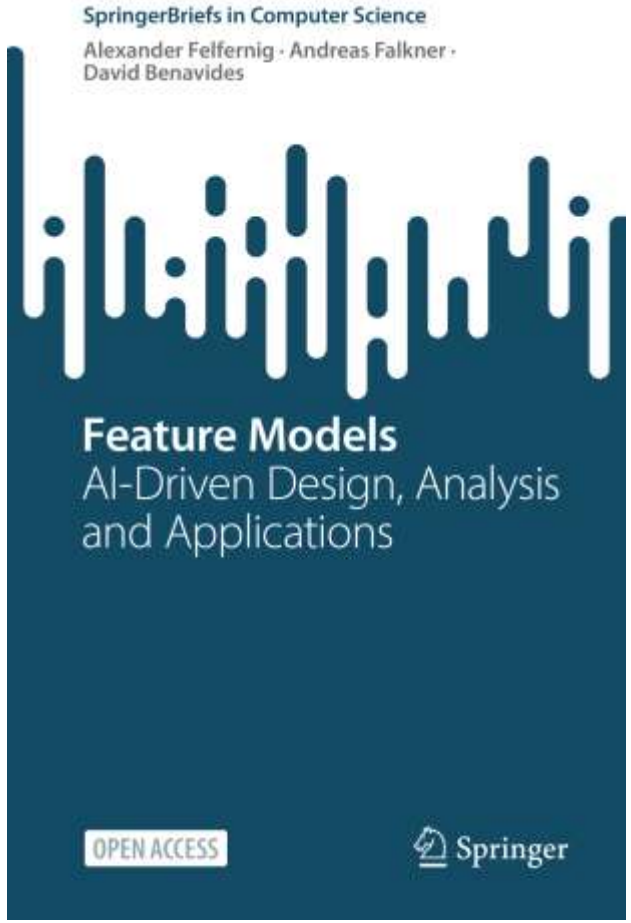
```
include
    Boolean.group-cardinality
    Arithmetic.aggregate-function
    Arithmetic.feature-cardinality
    Type

features
    Sandwich
        mandatory
            Bread {Calories 100, Sugar 20}
        optional
            Sauce
                or
                    Ketchup {Calories 40, Sugar 35}
                    Mustard {Calories 25, Sugar 5}
            Cheese
                [0..2] // Group cardinality
                    Cheddar {Calories 60}
                    Gouda {Calories 50}
                    Goat {Calories 35}
            Pickle cardinality [1..3] // Feature cardinality

constraints
    Ketchup => Cheese
    Bread.Sugar + Ketchup.Sugar + Mustard.Sugar < 60 // Attribute constraints
    sum(Calories) < 160 // Attribute aggregate
```

- **UVL**
- Fama XML
- Feature IDE
- Glencoe
- Others
  - CNF, JSON, …

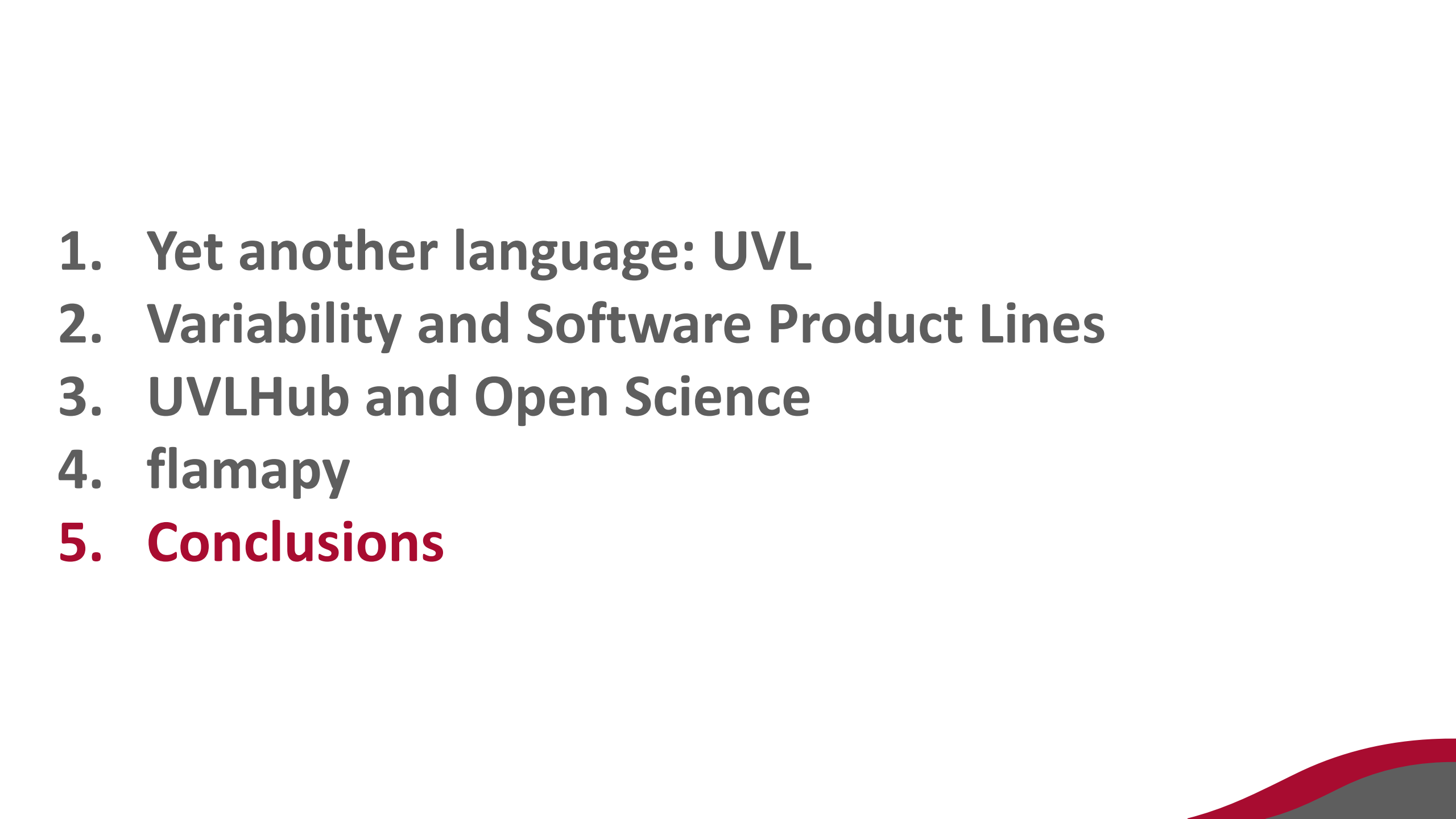
4. flamapy

Analysis operations

Atomic Sets	Average Branching Factor	Commonality	Configurations
Number of Configurations	Conflict Detection	Core Features	Count Leafs
Dead Features	Estimated Number of Configurations	False Optional Features	Feature Ancestors
Filter	Leaf Features	Max Depth	Satisfiable
	Satisfiable Configuration	Unique Features	



1. Yet another language: UVL
2. Variability and Software Product Lines
3. UVLHub and Open Science
4. flamapy
5. **Conclusions**



5. Conclusions

- **UVL** as a new textual language
- Using **UVLHub** for model sharing
- Analysing models with **flamapy**
- Following the **Open Science** approach

LangDev CON 2024

Open Science principles in software product lines: The case of the UVL ecosystem

David Romero-Organvidez, José A. Galindo, Megha Bhushan, Jose-Miguel Horcas, David Benavides