TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Forestry, Geosciences and Hydro sciences, Department of Geosciences, Geoinformation Systems

Moving Code in SDI: Sharing Geoprocessing Tools with Web Services

Matthias Müller, Lars Bernard, Johannes Brauner

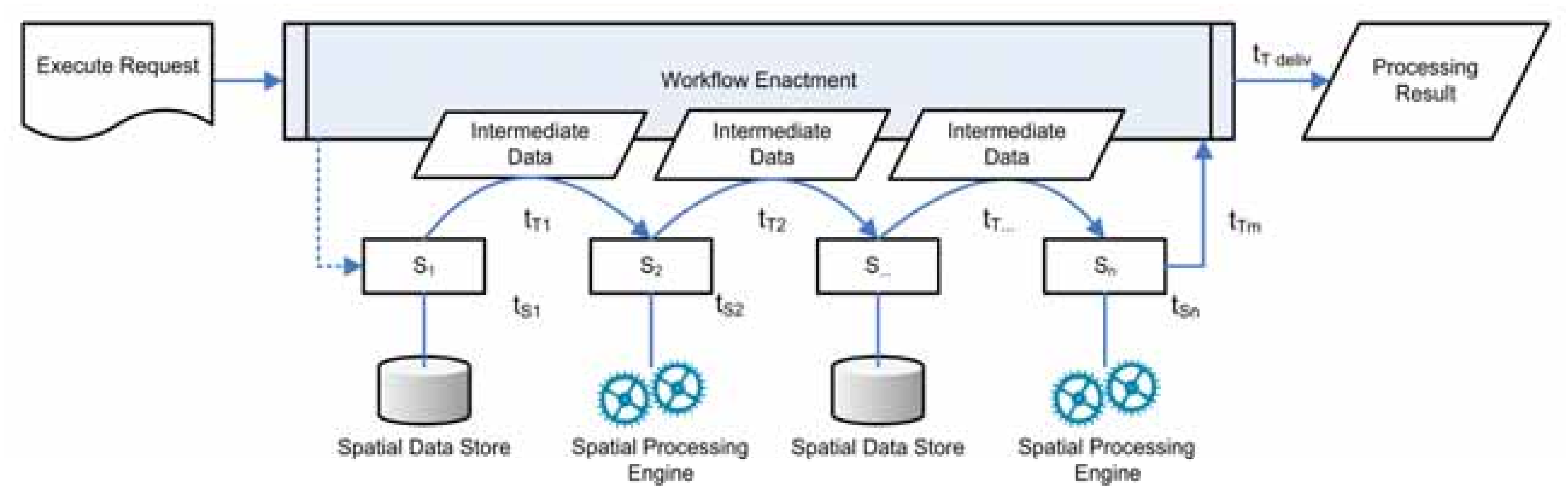
Outline

- Service-based geoprocessing
- “Moving Code” approaches
- “Moving Code” infrastructure & requirements
- Advantages in lifecycle management
- Assessment and outlook

Service-based Geoprocessing (GP)

- Data-driven (state-of-the-art)
  - The focus is set on data
  - GP Services provide a static set of spatial operations
  - Data is shipped around
- Code-driven
  - The focus is set on code
  - GP Services are supplied ad-hoc with new operations
  - GP Services receive dynamic updates
  - GP Services retrieve algorithms from different producers and locations
  - Code-driven infrastructures require “moving code” mechanisms

Data-driven Geoprocessing in SDI

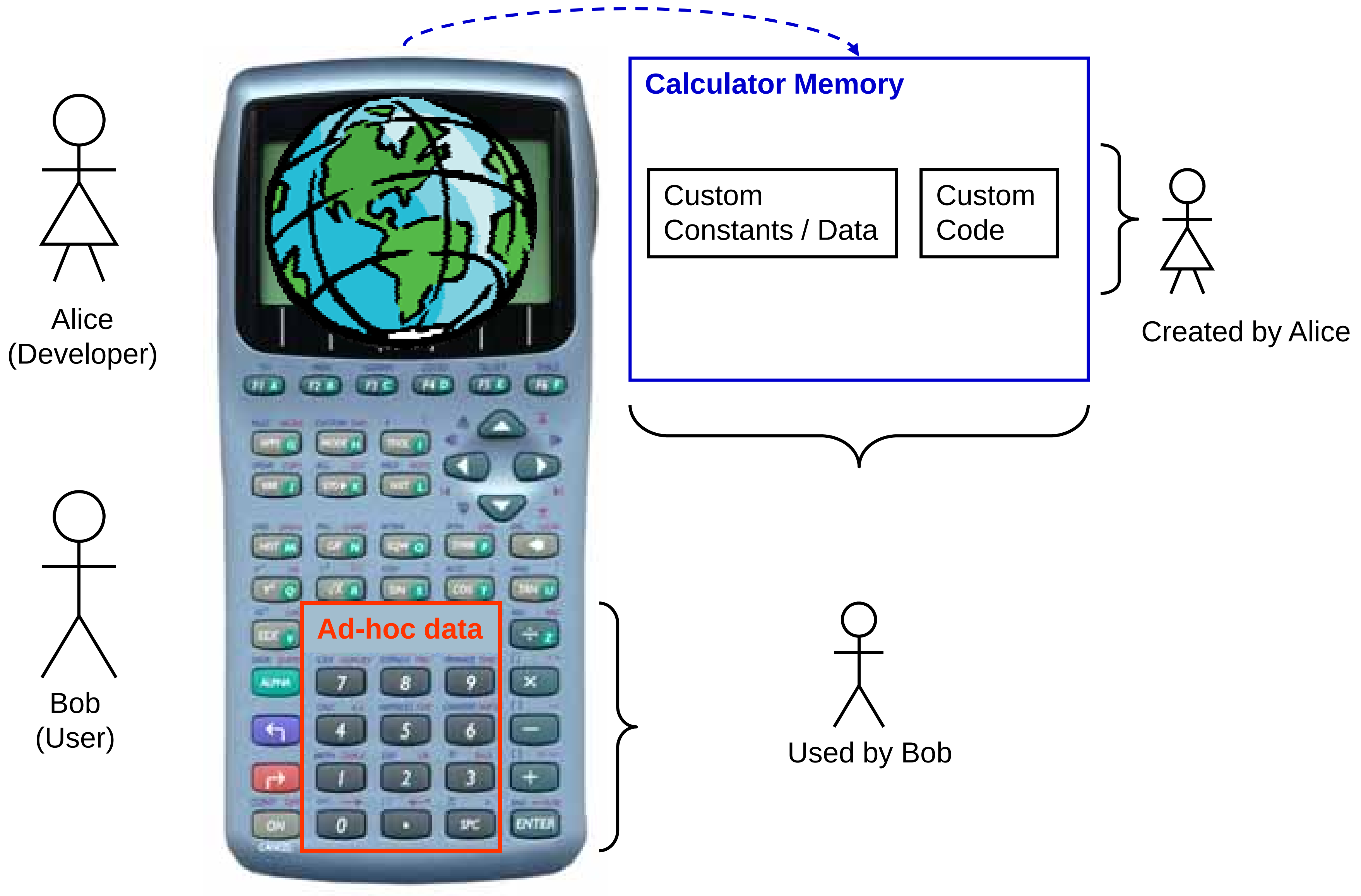


S – Service

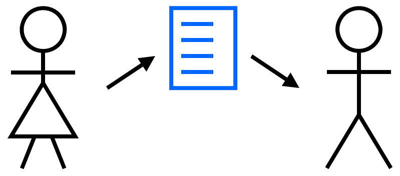
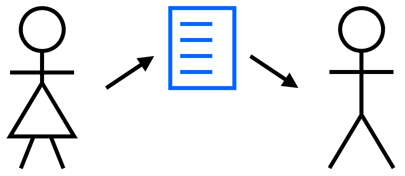
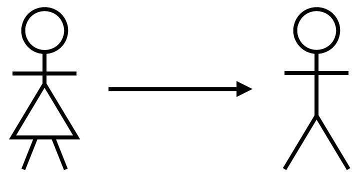
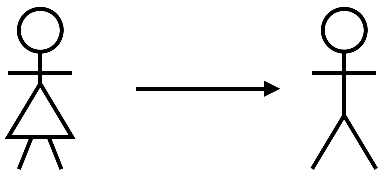
t_T – data transportation time

t_S – time for Web Service invocation

The “Code driven” Geospatial Calculator



“Moving Code” for Geoprocessing – A Classification

		Data Coupling	
		Tightly coupled	Loosely coupled
Execution Scheme	Instant execution	 Execute(Code)	 Execute(Code, Data)
	Permanent deployment	 Deploy(Code) Execute() [multiple]	 Deploy(Code) Execute(Data) [multiple]

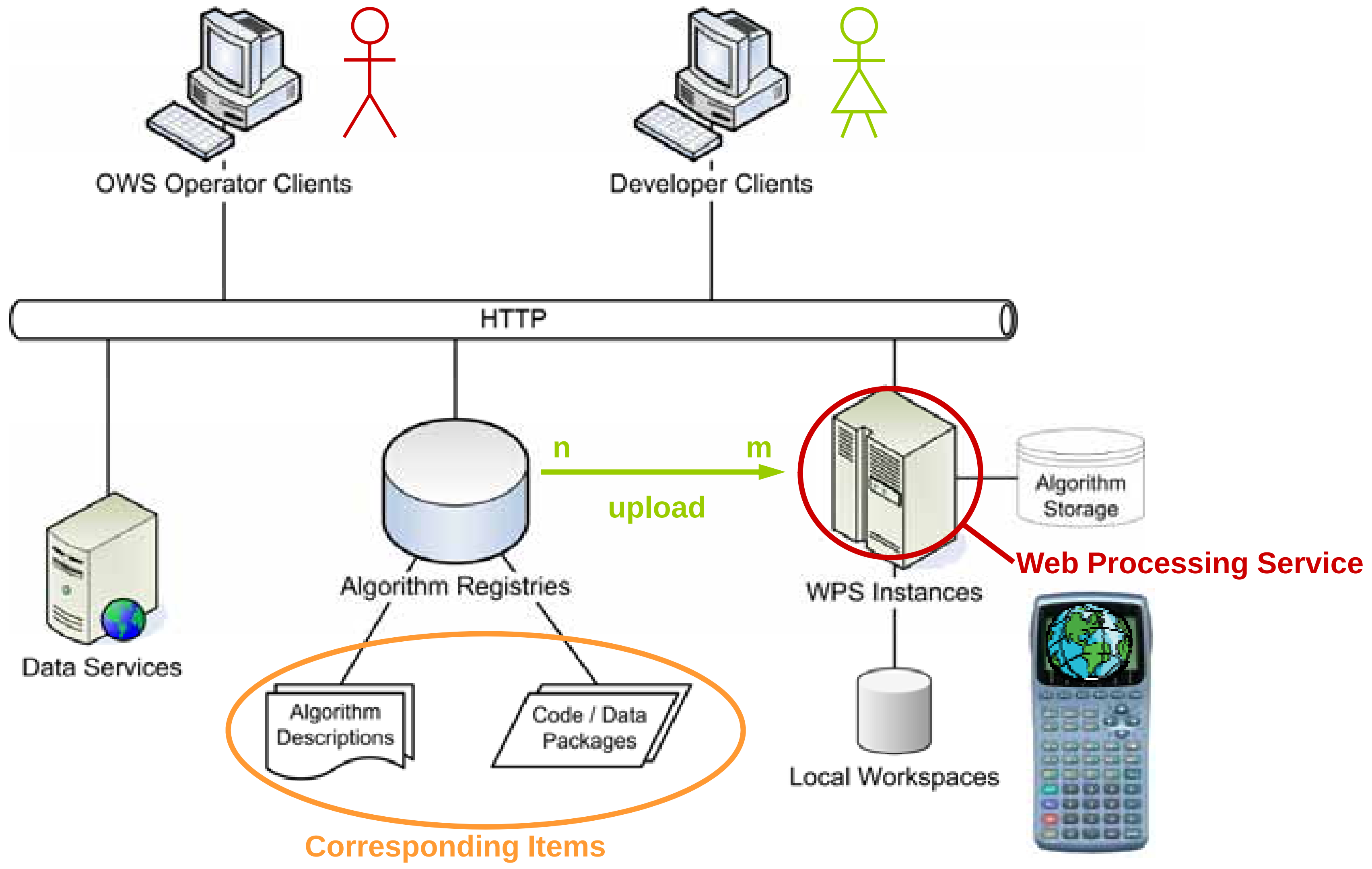
“Moving Code” for Geoprocessing – A Classification

Data Coupling

Execution Scheme		Tightly coupled	Loosely coupled
	Instant execution	▪ Web Coverage Processing Service ▪ Filter Encoding ▪ SQL statements	▪ Filter Encoding
	Permanent deployment	▪ Stored SQL query ▪ (Transactional WPS)	▪ (Transactional WPS)

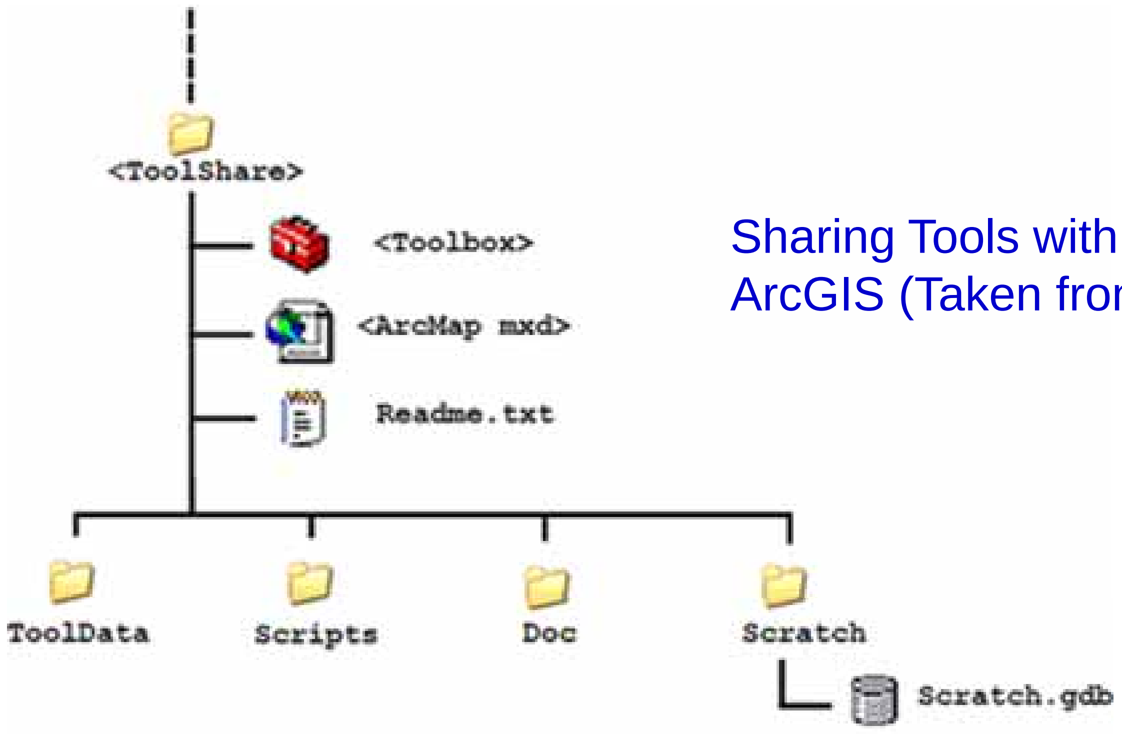
- Grid infrastructures

A Simple “Moving Code” Architecture



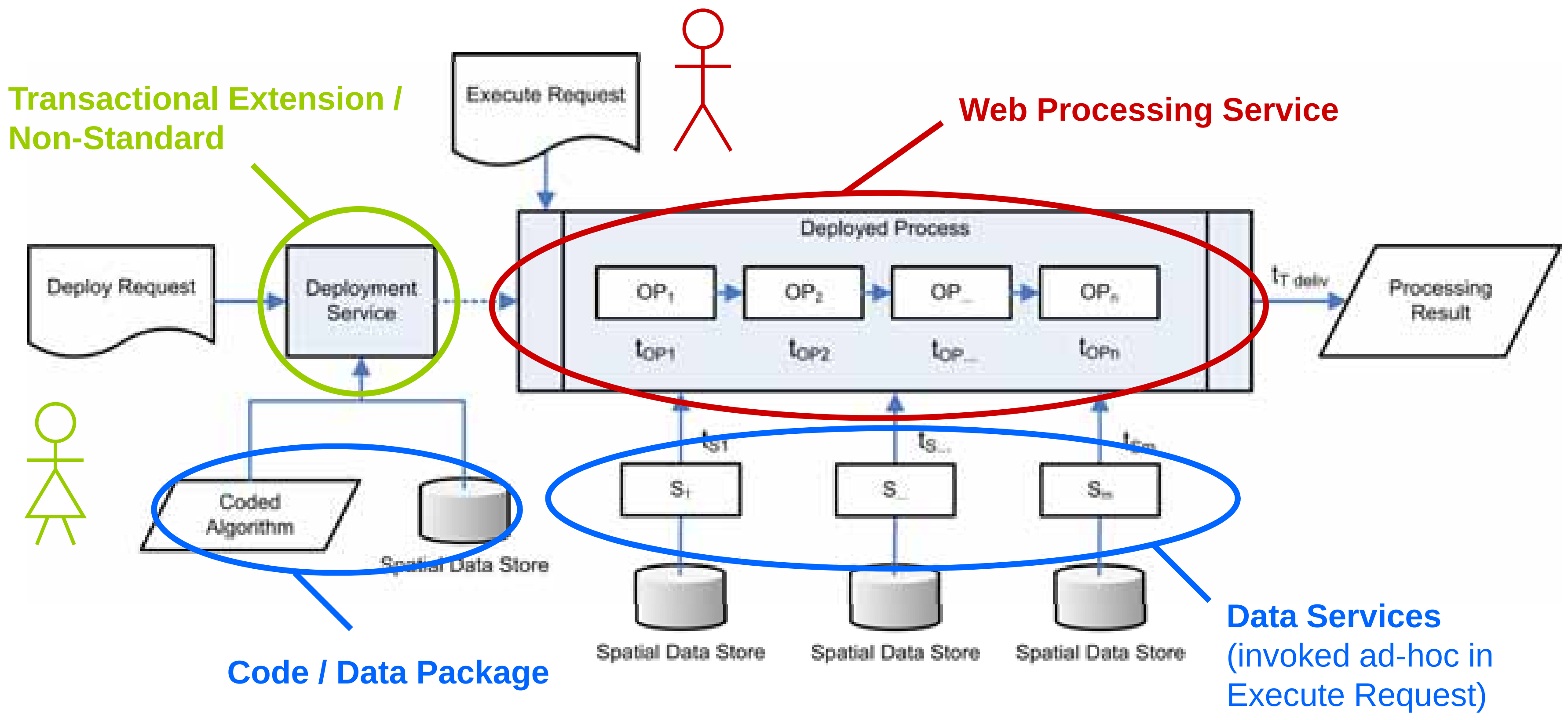
Workspaces to Cache Tightly Coupled Data

- Workspaces contain code and some (constant) data
- No need to ship data around that is required in every execution ("caching")
- Workspaces have to be designed to run on each intended platform

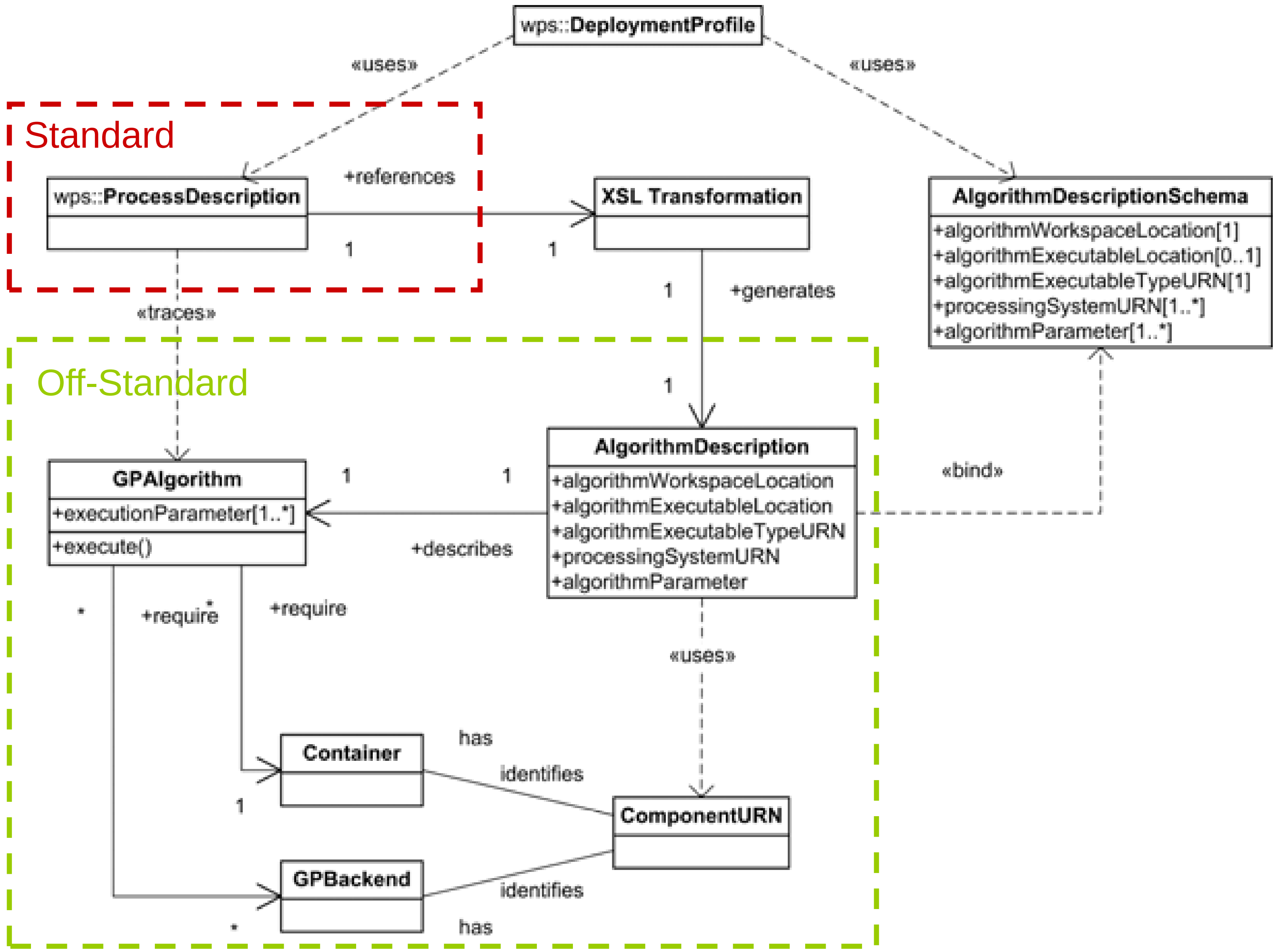


Sharing Tools with Workspaces in ArcGIS (Taken from the ESRI Help)

Deployment & Execution

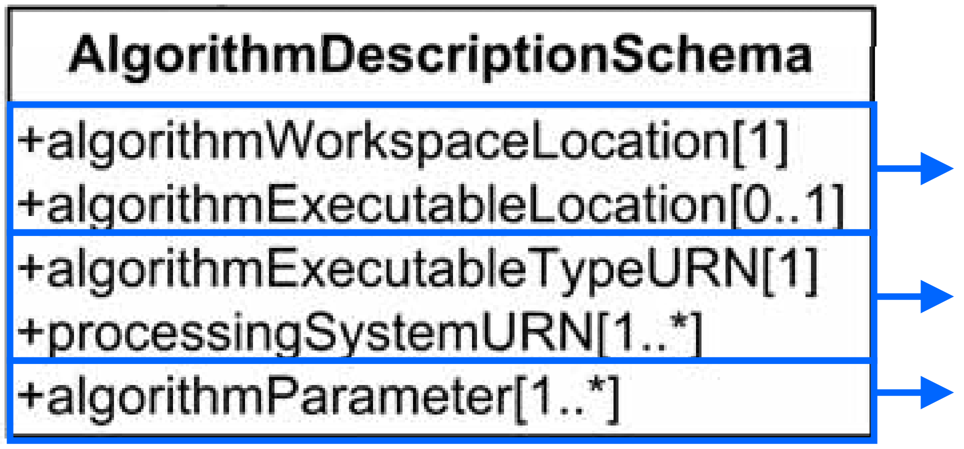


Linking WPS Process Interfaces, Algorithms and Workspaces



Linking WPS Process Interfaces, Algorithms and Workspaces

- We lack a generic Geoprocessing algebra …
  - How to communicate the Algorithm and the Workspace to the Service?
  - The number of Geoprocessing Systems is limited …
  - Can we instead describe the required runtime environment?
  - What about the parameter mapping?

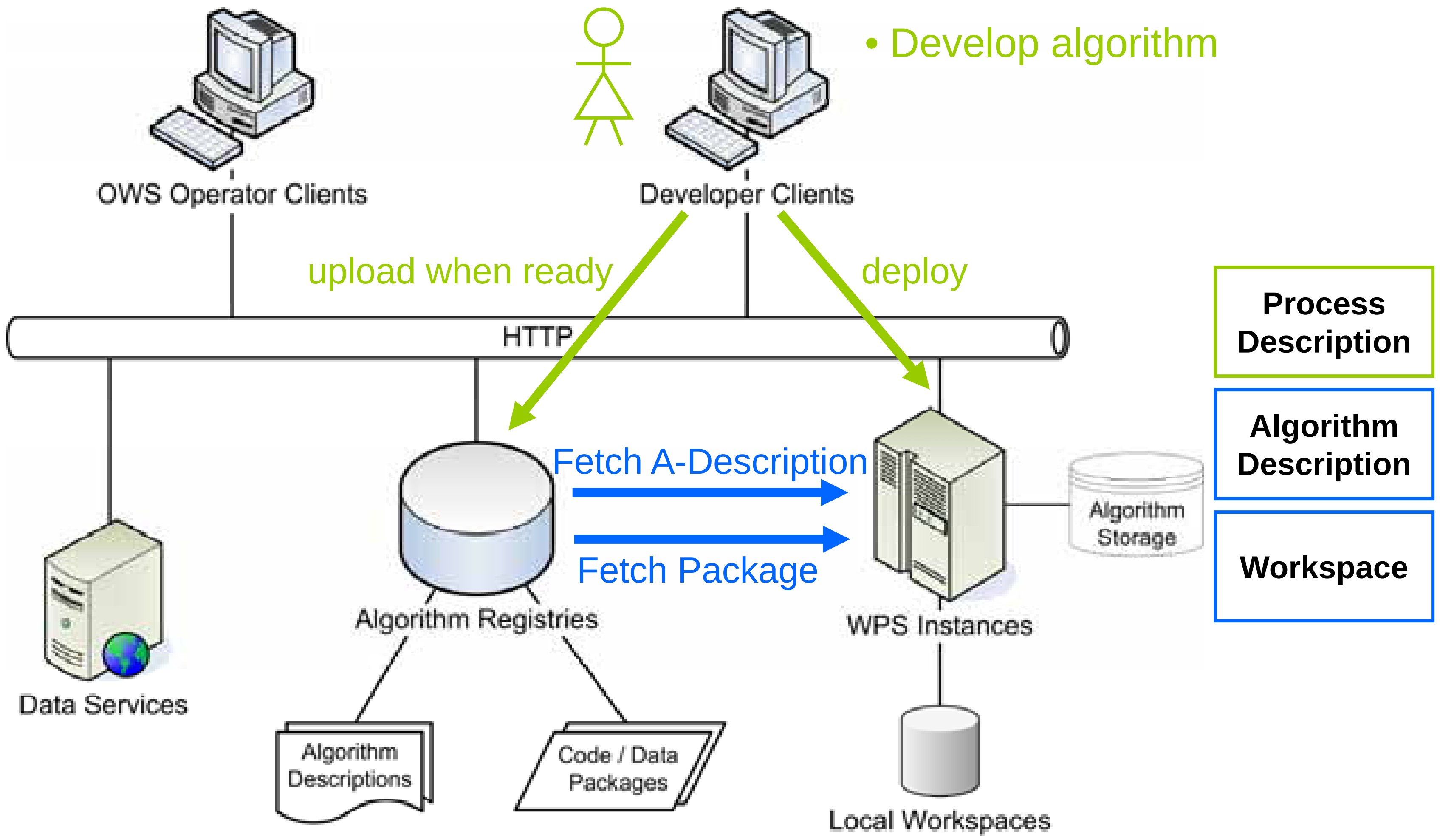

AlgorithmDescriptionSchema
+algorithmWorkspaceLocation[1]
+algorithmExecutableLocation[0..1]
+algorithmExecutableTypeURN[1]
+processingSystemURN[1..*]
+algorithmParameter[1..*]

Where to find the Code / Data Package and the executable.

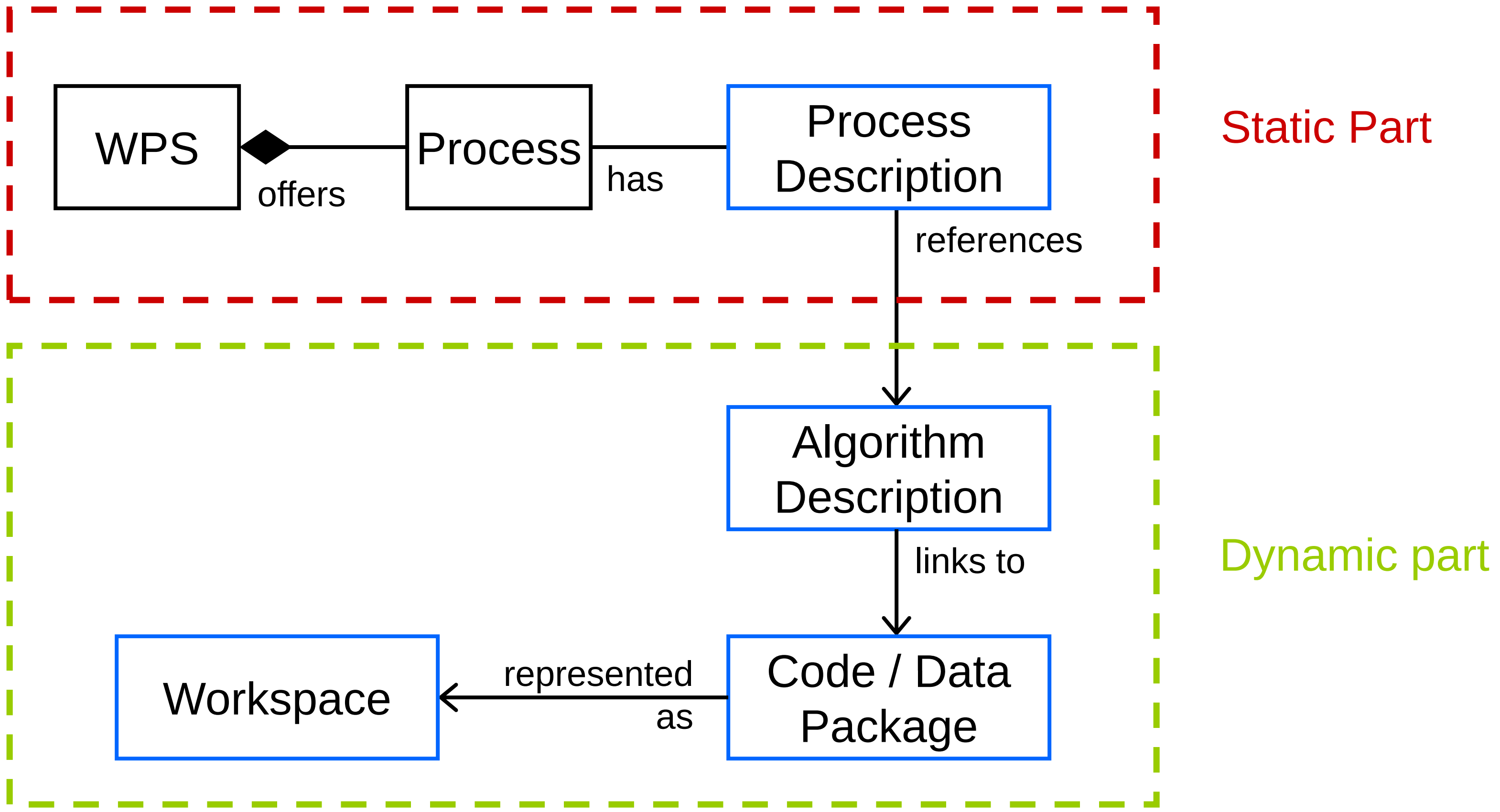
The required runtime environment.

The legacy parameters.

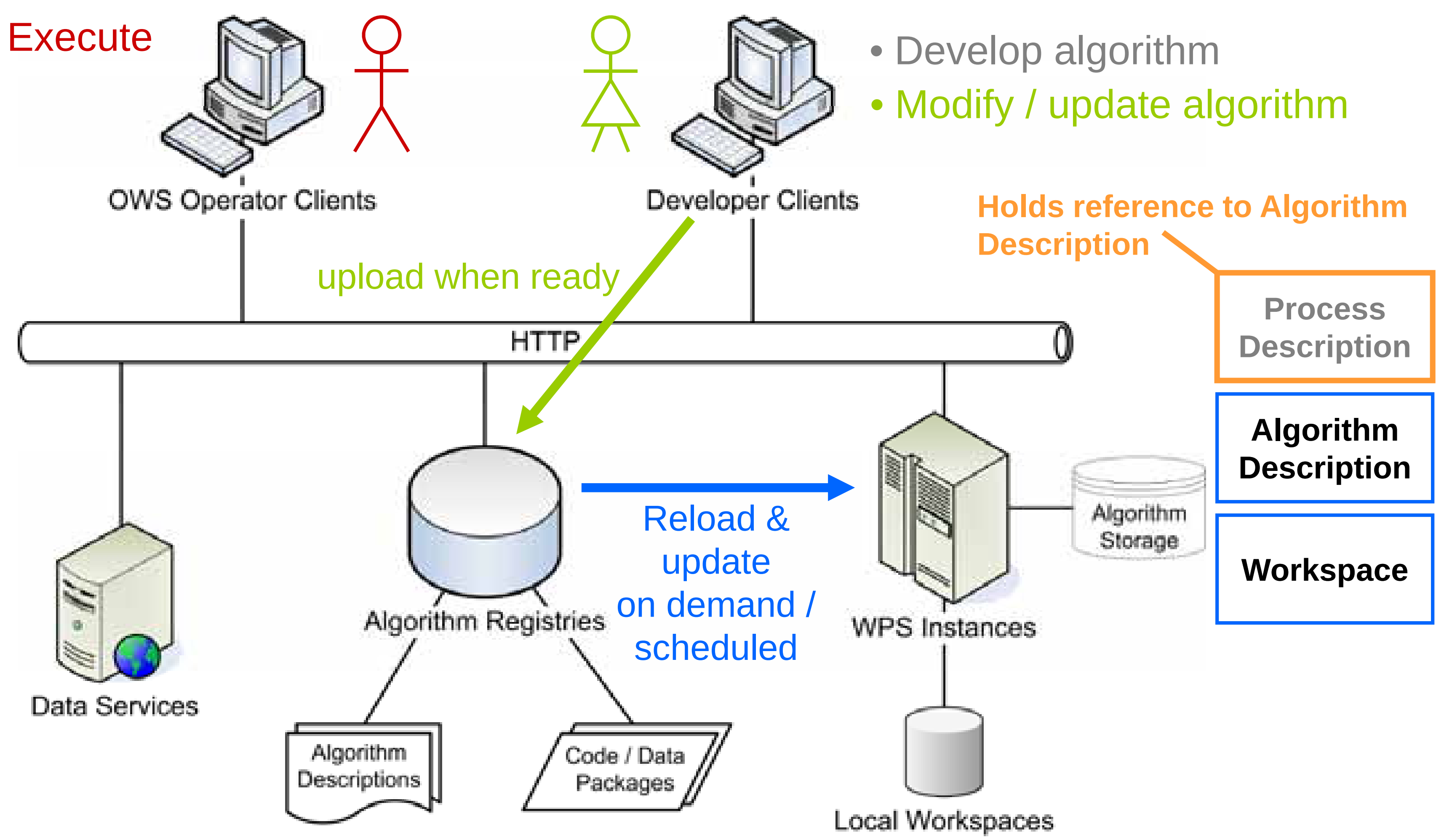
Deployment


• Develop algorithm
OWS Operator Clients
Developer Clients
upload when ready
deploy
HTTP
Process Description
Algorithm Description
Workspace
Fetch A-Description
Fetch Package
Algorithm Storage
Algorithm Registries
WPS Instances
Data Services
Algorithm Descriptions
Code / Data Packages
Local Workspaces

Life Cycle Management

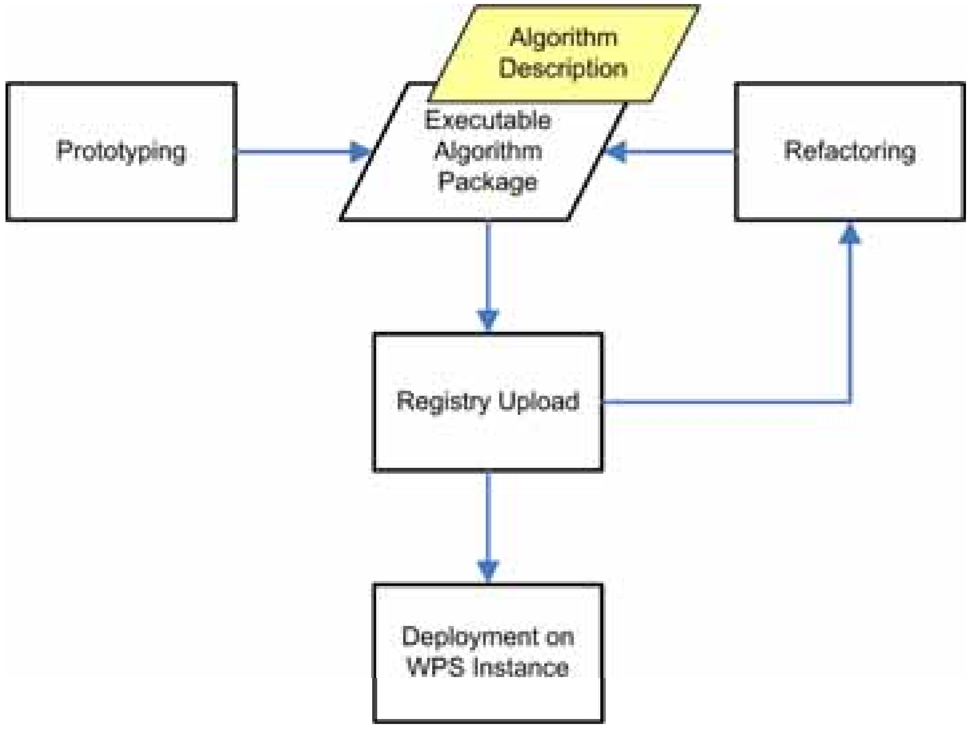


Life Cycle Management



Life Cycle Management

- Develop algorithm
- Modify / update algorithm



Algorithm Storage

Process Description

Algorithm Description

Workspace

Code-driven vs. Data-driven

- ‘Moving code’ approaches are beneficial if:
  - Algorithms are frequently changed and evolved
  - Identical algorithms have to be deployed at or shared among several service instances
  - A substantial amount of data can be shipped with the algorithm and stored prior to execution
  - Some tightly coupled data sets can be used to increase performance (caching impact)
  - Algorithms have to be placed at a processing service that resides “close” to the data (bandwidth impact)

Code-driven vs. Data-driven

- Data-driven approaches are beneficial if:
  - All required data sets change frequently
  - One-time assembly and execution of workflows
  - Real-time response for complex service chains is not required
  - The required atomic operators are available at processing services or
  - The required simple operators are available at the data service level

What’s next?

- Progress in the design of well defined geoprocessing algebras
- Evaluate potential for parallelization
- Create Service Grids
- Evolve the Standards