

UNITE

An Adapter for Transforming Analysis Tools to Web Services via OSLC



Bohuslav Křena

Aleš Smrčka

Tomáš Vojnar

Ondřej Vašíček

Jan Fiedor

Honeywell

Tomáš Kratochvíla

@ESEC/FSE 2022,
Singapore



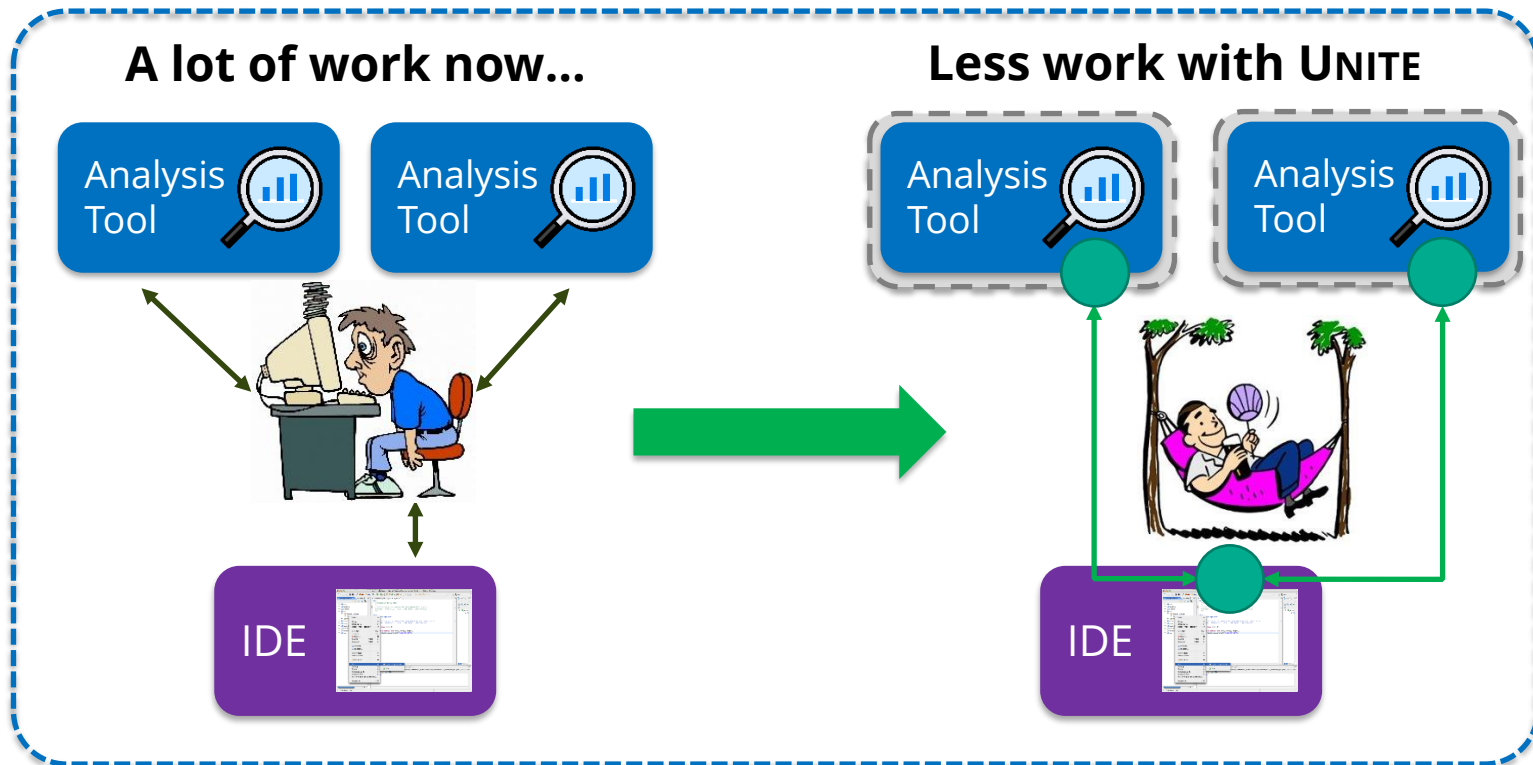
Goal: Making analysis* **tools easier to adopt and use**

* static / dynamic analysis, verification, ... almost any command-line tool

Goal: Making analysis* tools easier to adopt and use

* static / dynamic analysis, verification, ... almost any command-line tool

Approach: Transforming analysis tools to web services
and using them through convenient clients



How to adopt a new command-line analysis tool?

*How to adopt a **new command-line analysis tool**?*

a) Using it **on your PC**

- **Complicated setup** (OS, incompatibility, dependencies, ...)
 - **Colleagues** need to go through the same process **again**
- **Limited execution** (insufficient hardware, long run duration, PC under load, ...)

How to adopt a new command-line analysis tool?

a) Using it **on your PC**

- **Complicated setup** (OS, incompatibility, dependencies, ...)
 - **Colleagues** need to go through the same process **again**
- **Limited execution** (insufficient hardware, long run duration, PC under load, ...)

b) Use it **remotely on a server** using SSH

- Solves the **PC issues**
 - Setup – tailored **dedicated server**, only **setup once**
 - Execution – **offloaded** to a **higher performance** server, **remote** access
- **Extra work** (transferring files to/from the server, monitoring execution, ...)
- **Bad user-experience***, could be too **complicated for some** users
(* not everyone is a programmer)

- a) Using it **on your PC**
- b) Use it **remotely on a server** using SSH

- a) Using it **on your PC**
- b) Use it **remotely on a server** using SS

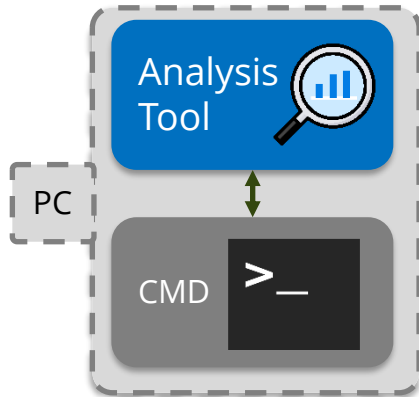
- c) Using it **as a web service** (from scratch)
 - Can be used through a **user-friendly client**
 - But need to **create the web service and the client** (a lot of work)

- a) Using it **on your PC**
- b) Use it **remotely on a server** using SS

- c) Using it **as a web service** (from scratch)
 - Can be used through a **user-friendly client**
 - But need to **create the web service and the client** (a lot of work)

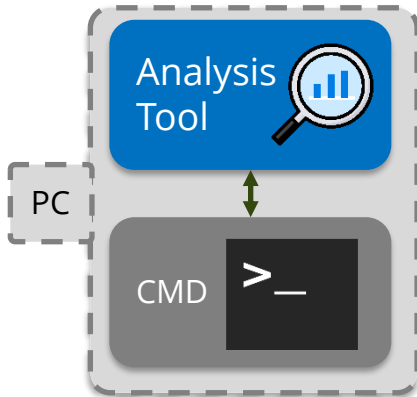
- d) Using it as a web service **with UNITE**
 - **Transform** the tool to a web service almost **for free**
 - **No source code modifications** required (analysis tool, nor UNITE)
 - **Don't have to be an expert** on web services
 - Need to create **configuration files only**
 - Uses a **standardized interface** (easier integration & client reuse)

Without UNITE

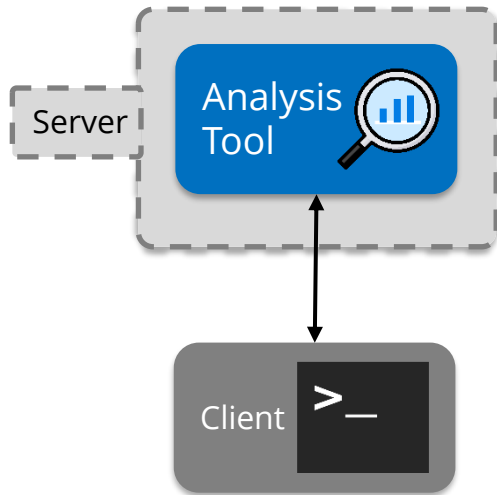


Using a tool through its
CMD interface

Without UNITE

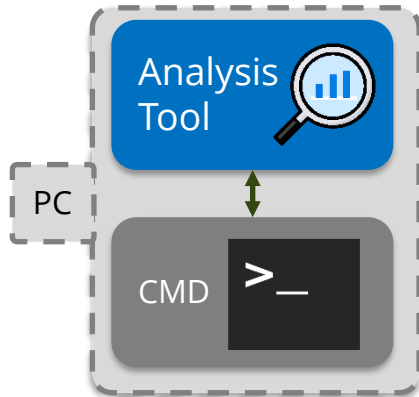


Using a tool through its
CMD interface

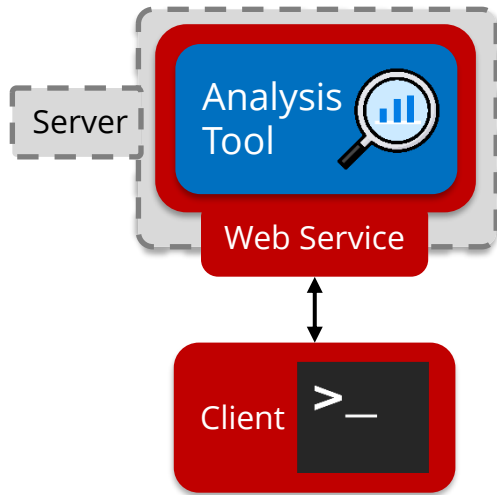


Using a tool through your
own **custom web service**

Without UNITE



Using a tool through its
CMD interface

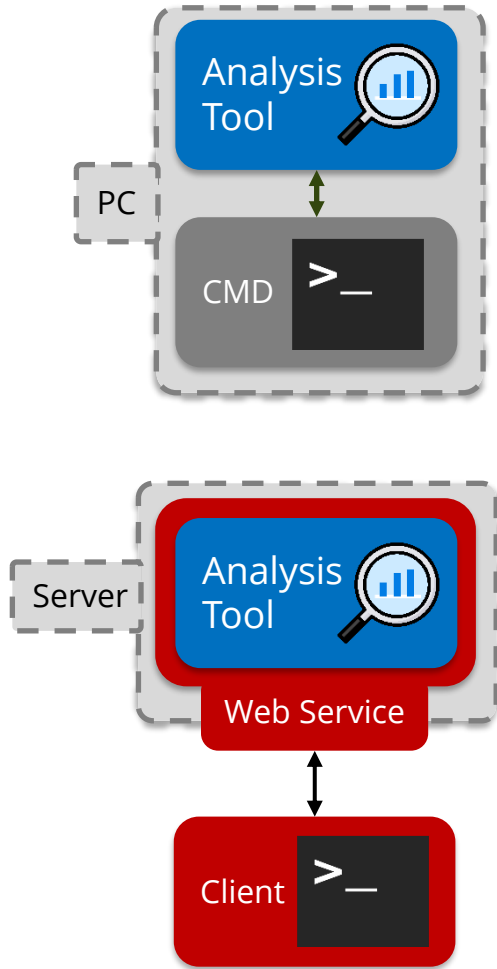


Using a tool through your
own **custom web service**

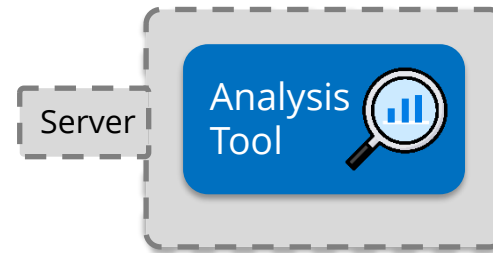


(A lot of work)

Without UNITE

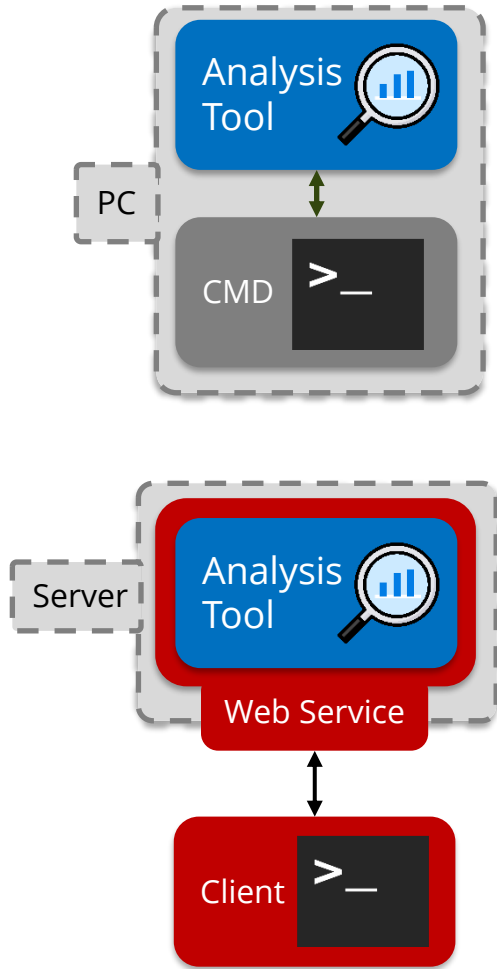


With UNITE

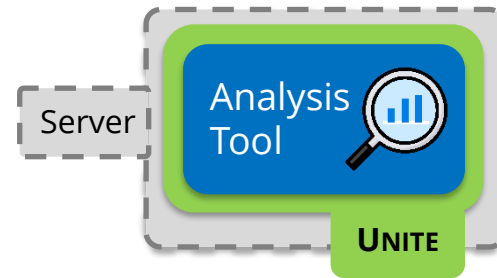


Using a tool remotely on a server with **UNITE**

Without UNITE



With UNITE



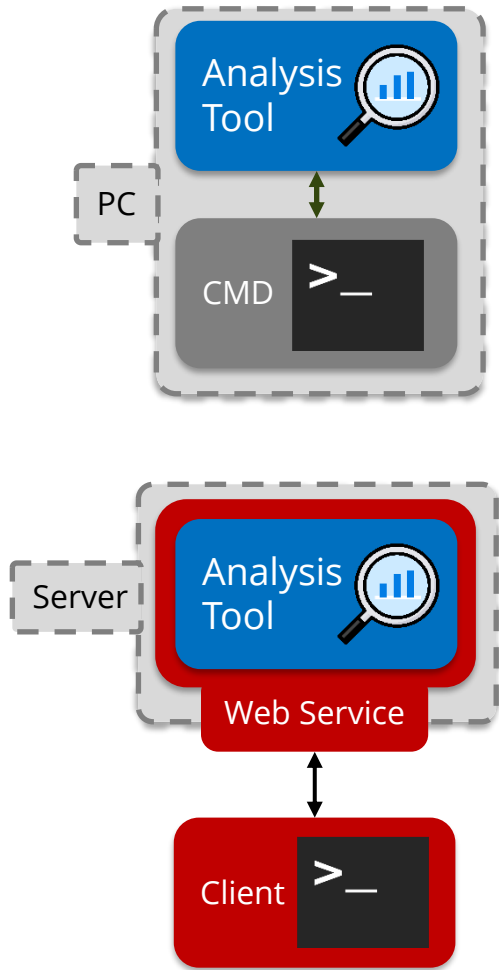
UNiversal Analysis Adapter



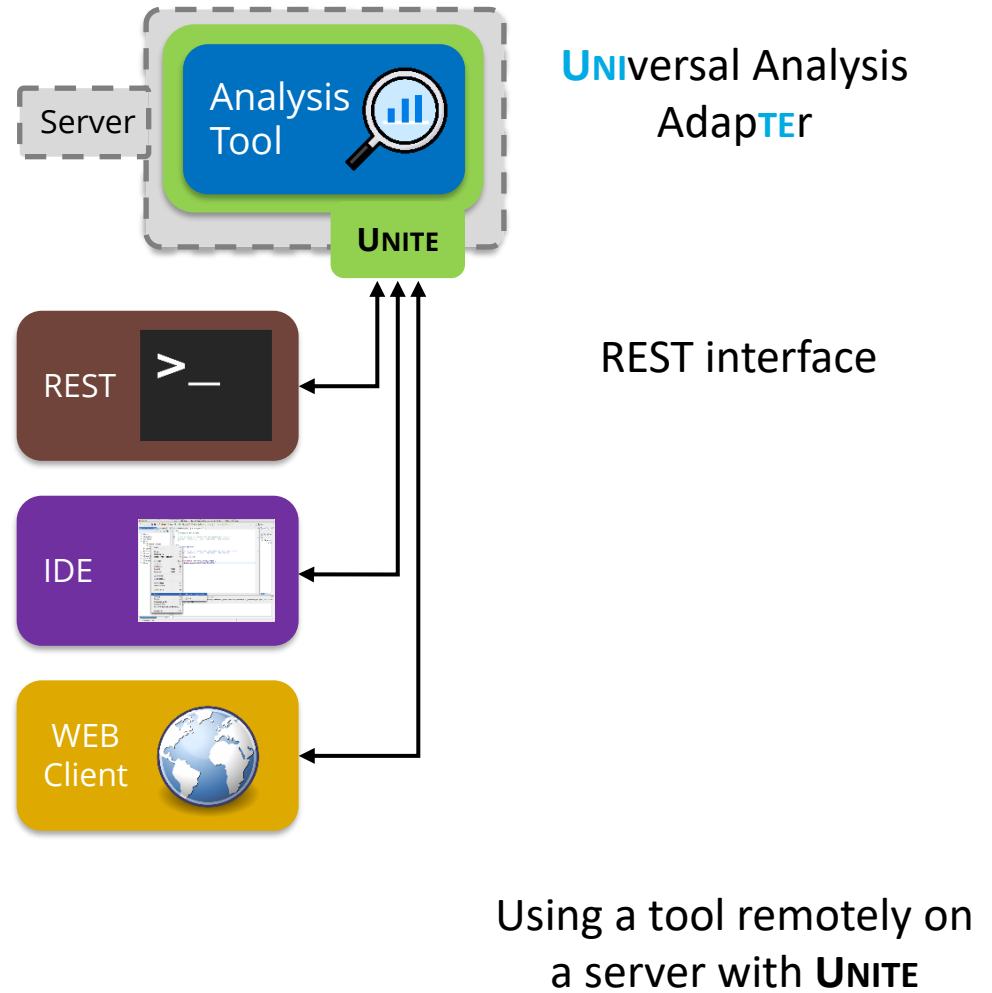
(A lot less work)

Using a tool remotely on a server with **UNITE**

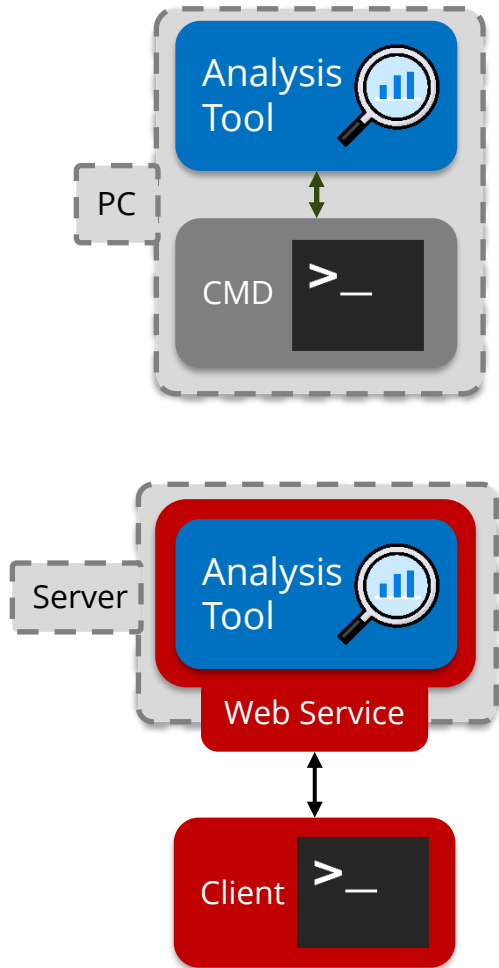
Without UNITE



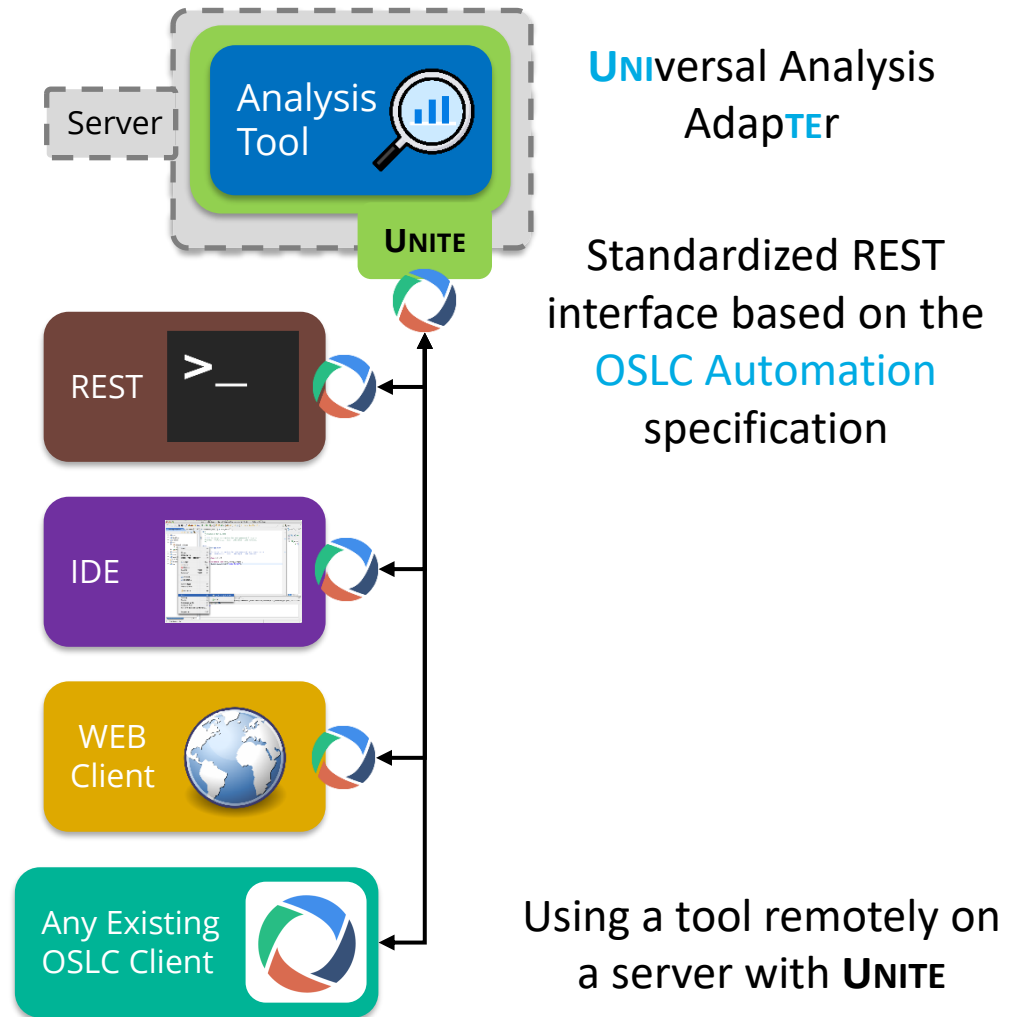
With UNITE



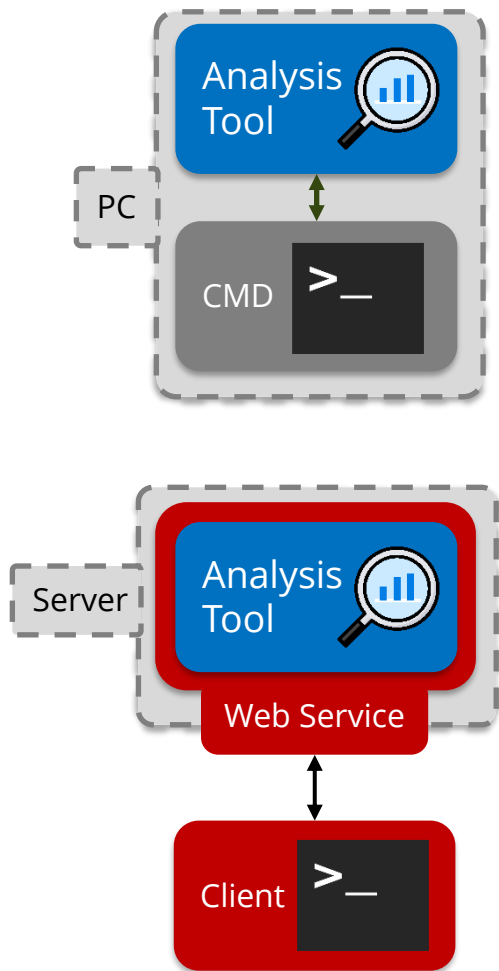
Without UNITE



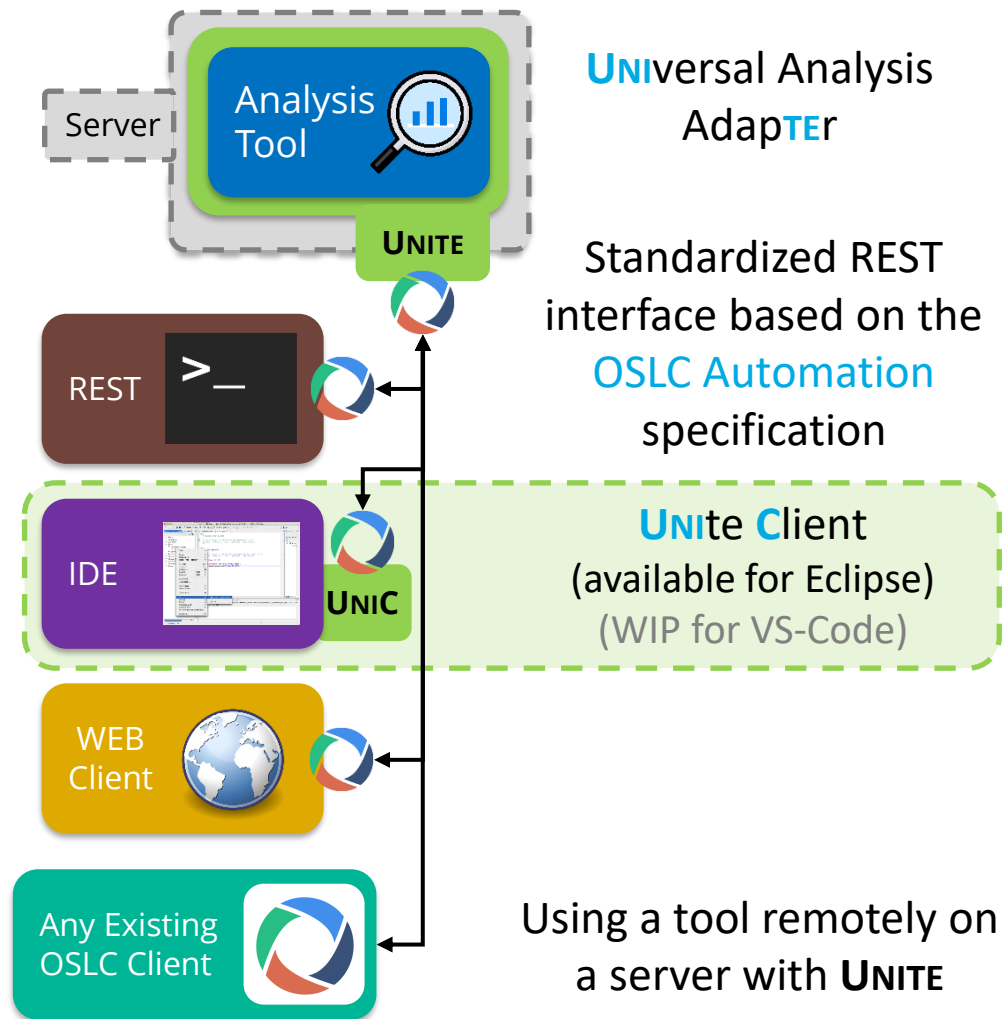
With UNITE



Without UNITE



With UNITE



The screenshot shows the Eclipse IDE interface. The Project Explorer on the left shows the project structure. The main editor displays the code for `Utils.java`. A yellow tooltip highlights a resource leak warning at line 242, stating: "Multiple markers at this line - Resource Leak: resource of type 'java.util.zip.ZipOutputStream' acquired to 'zipOutputStream' by call to 'new()' at line 219 is not released after line 242. - Resource Leak: resource of type 'java.io.FileInputStream' acquired to 'fileInStream' by call to 'FileInputStream(...)' at line 227 is not released after line 242."

The Problems view at the bottom shows a table of errors:

| Description | Resource | Path | Location | Type |
|---|--------------------------|-------------|----------|---------|
| Null Dereference: object returned by 'RequestRunnerQuee | RequestRunnerQueues.java | /shared/... | line 60 | Problem |
| Null Dereference: object 'paramDef' last assigned on lin | ExecutionParameter.java | /shared/... | line 87 | Problem |
| Resource Leak: resource of type 'java.io.FileInputStream' | Utils.java | /shared/... | line 242 | Problem |
| Resource Leak: resource of type 'java.util.zip.ZipOutputS | Utils.java | /shared/... | line 242 | Problem |
| Resource Leak: resource of type 'java.util.zip.ZipFile' acq | Utils.java | /shared/... | line 266 | Problem |

The screenshot shows the Eclipse IDE interface. The main editor displays the file `Utils.java` with the following code:

```

237 zipOutputStream.flush();
238 fileOutputStream.flush();
239 zipOutputStream.close();
240 fileOutputStream.close();
241
242
243
244
245
246
247
248 * @param zipFile
249 * @throws IOException
250 */
251 public static void unzipFile(Path dirToUnzipTo, File zipFile) throw
252 {
253     ZipFile zf = new ZipFile(zipFile);
254     Enumeration<? extends ZipEntry> zipEntries = zf.entries();

```

A tooltip is displayed over line 242, indicating multiple markers:

- Resource Leak: resource of type `java.util.zip.ZipOutputStream` acquired to `zipOutputStream` by call to `new()` at line 219 is not released after line 242.
- Resource Leak: resource of type `java.io.FileInputStream` acquired to `fileInStream` by call to `FileInputStream(...)` at line 227 is not released after line 242.

The Problems view at the bottom shows 5 errors:

| Description | Resource | Path | Location | Type |
|---|--------------------------|-------------|----------|---------|
| Null Dereference: object returned by 'RequestRunnerQue | RequestRunnerQueues.java | /shared/... | line 60 | Problem |
| Null Dereference: object 'paramDef' last assigned on lin | ExecutionParameter.java | /shared/... | line 87 | Problem |
| Resource Leak: resource of type 'java.io.FileInputStream' | Utils.java | /shared/... | line 242 | Problem |
| Resource Leak: resource of type 'java.util.zip.ZipOutputS | Utils.java | /shared/... | line 242 | Problem |
| Resource Leak: resource of type 'java.util.zip.ZipFile' acq | Utils.java | /shared/... | line 266 | Problem |

The Tasks view shows a task: **Analyse with Facebook Infer (BUT server)**. A red box highlights the number **1** in the left sidebar and the task name.

The screenshot shows the Eclipse IDE interface. The top part displays the Project Explorer on the left and the code editor for `Utils.java` on the right. The code editor shows lines 237 to 254, with a yellow tooltip highlighting a resource leak warning at line 242. The tooltip text is:

```

Multiple markers at this line
- Resource Leak: resource of type `java.util.zip.ZipOutputStream` acquired to
`zipOutputStream` by call to `new()` at line 219 is not released after line 242.
- Resource Leak: resource of type `java.io.FileInputStream` acquired to `fileInStream`
by call to `FileInputStream(...)` at line 227 is not released after line 242.
    
```

The bottom part of the screenshot shows the Problems view, which is highlighted with a red box. It displays a table of errors:

| Description | Resource | Path | Location | Type |
|---|--------------------------|-------------|----------|---------|
| Null Dereference: object returned by `RequestRunnerQue | RequestRunnerQueues.java | /shared/... | line 60 | Problem |
| Null Dereference: object `paramDef` last assigned on lin | ExecutionParameter.java | /shared/... | line 87 | Problem |
| Resource Leak: resource of type `java.io.FileInputStream` | Utils.java | /shared/... | line 242 | Problem |
| Resource Leak: resource of type `java.util.zip.ZipOutputS | Utils.java | /shared/... | line 242 | Problem |
| Resource Leak: resource of type `java.util.zip.ZipFile` acq | Utils.java | /shared/... | line 266 | Problem |

Two red boxes with numbers '1' and '2' are overlaid on the image. Box '1' is on the 'Tasks' menu item in the Project Explorer, and box '2' is on the 'Problems' view tab.

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure with 'src/main/java' selected.
- Code Editor:** Displays the source code for 'Utils.java'. Lines 237-241 show flush and close operations. Line 242 has a warning: "Multiple markers at this line". Lines 243-246 show resource leak warnings for 'zipOutputStream' and 'fileInStream'.
- Problems View:** Shows 5 errors and 10 warnings. The errors are:

| Description | Resource | Path | Location | Type |
|---|--------------------------|-------------|----------|---------|
| Null Dereference: object returned by 'RequestRunnerQuee | RequestRunnerQueues.java | /shared/... | line 60 | Problem |
| Null Dereference: object 'paramDef' last assigned on lin | ExecutionParameter.java | /shared/... | line 87 | Problem |
| Resource Leak: resource of type 'java.io.FileInputStream' | Utils.java | /shared/... | line 242 | Problem |
| Resource Leak: resource of type 'java.util.zip.ZipOutputS | Utils.java | /shared/... | line 242 | Problem |
| Resource Leak: resource of type 'java.util.zip.ZipFile' acq | Utils.java | /shared/... | line 266 | Problem |

- OASIS open project that defines **specifications** for integrating tools across the entire development lifecycle
- Interfaces are modeled based on **domains**
 - Made up of **resources** (such as test cases, issues, users,...)
 - Each resource has a unique **URI**
 - e.g., Quality Management, Requirements Management, **Automation**, ...
- OSLC **participants**:
 - Providers – **servers** X Consumers – **clients**
 - Use self-describing **RESTful** APIs, **HTTP**, serialized resource representation (RDF, XML, Turtle, or JSON)
- **Advantages** of OSLC
 - Open-source, distributed, support for data from various domains, flexible and extensible, traceability, service discovery, tooling support (Eclipse Lyo), ...



- UNITE is an [OSLC Automation provider](#) (server)

- UNITE is an OSLC Automation provider (server)

OSLC Automation domain:

Automation Plans

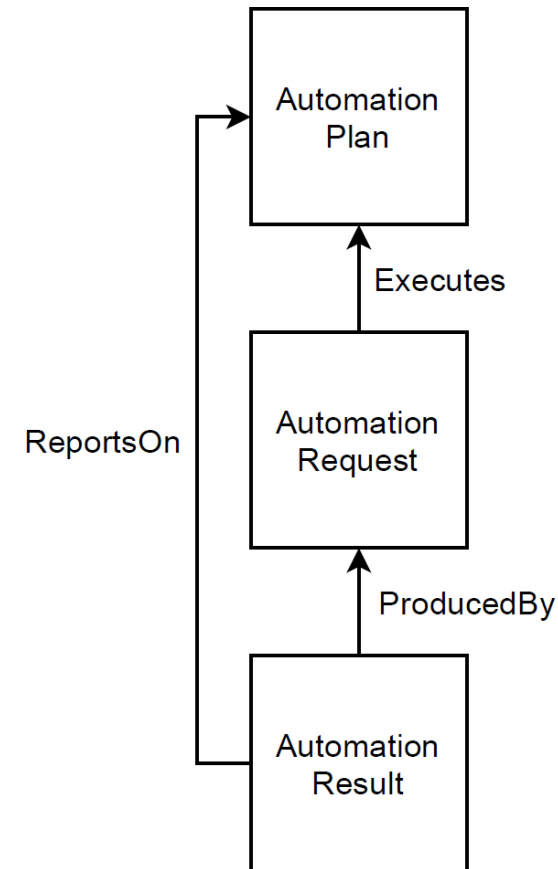
- Available units of automation and their parameters

Automation Request

- Requests to execute an Automation Plan

Automation Results

- Outputs of the execution



- UNITE is an **OSLC Automation provider** (server)

OSLC Automation domain:

Automation Plans

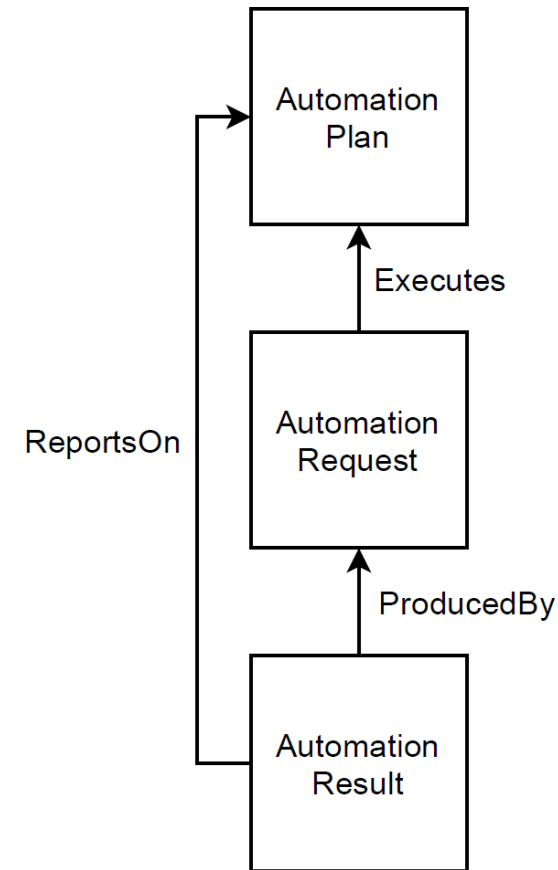
- Available units of automation and their parameters

Automation Request

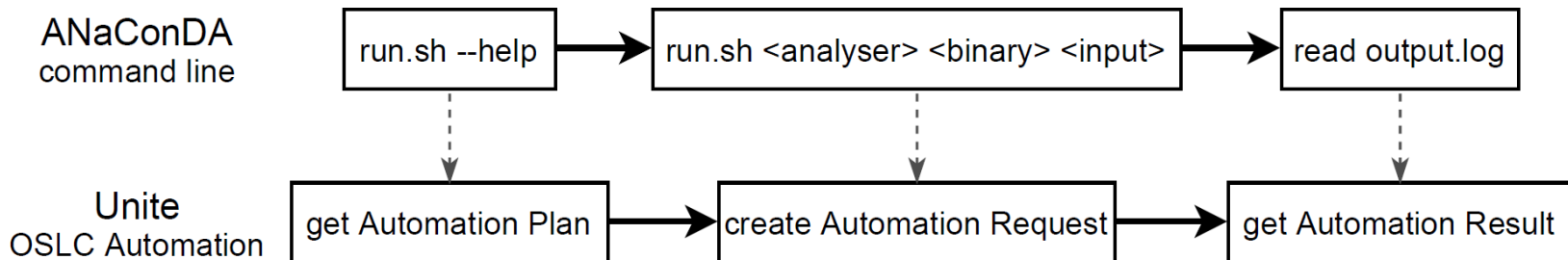
- Requests to execute an Automation Plan

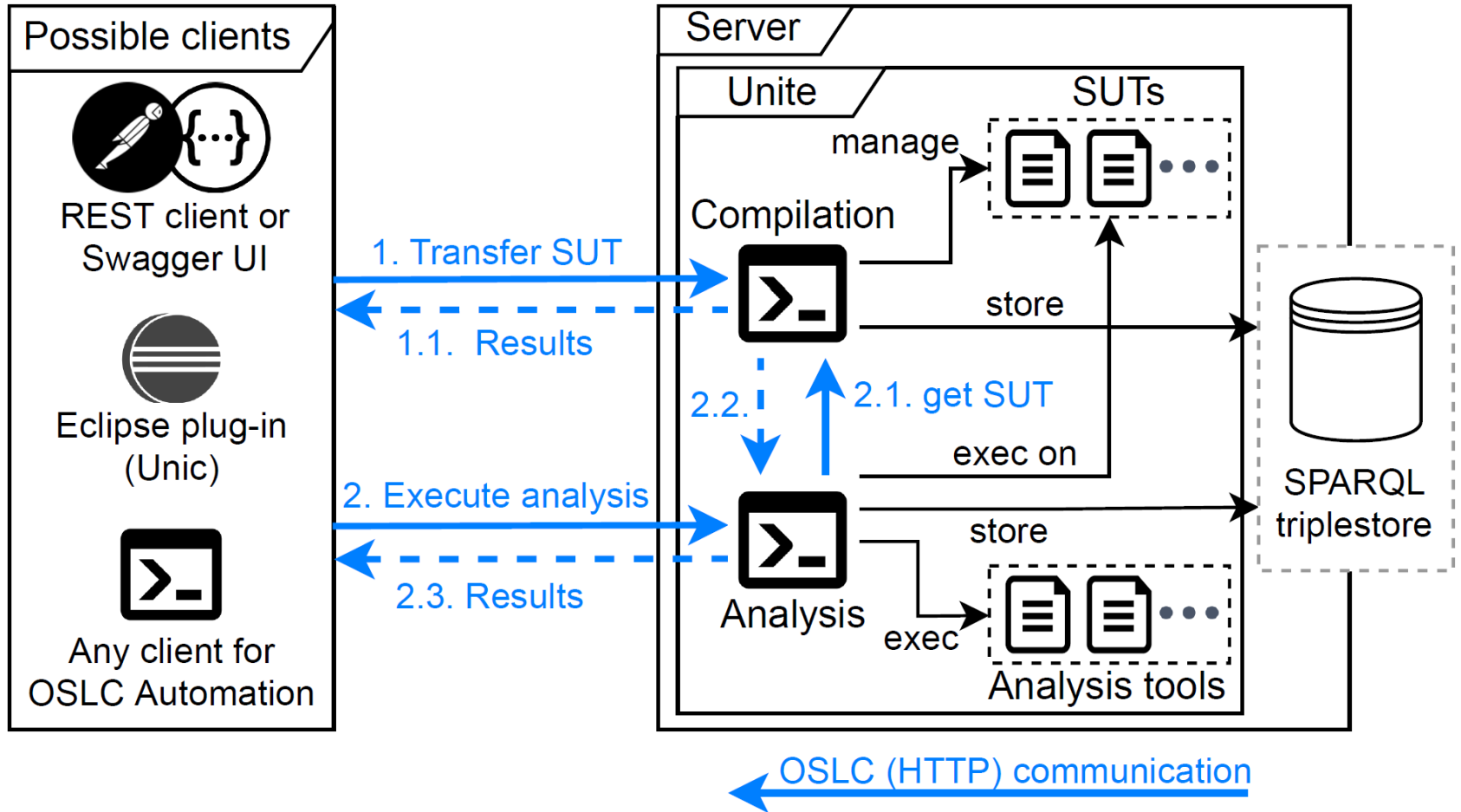
Automation Results

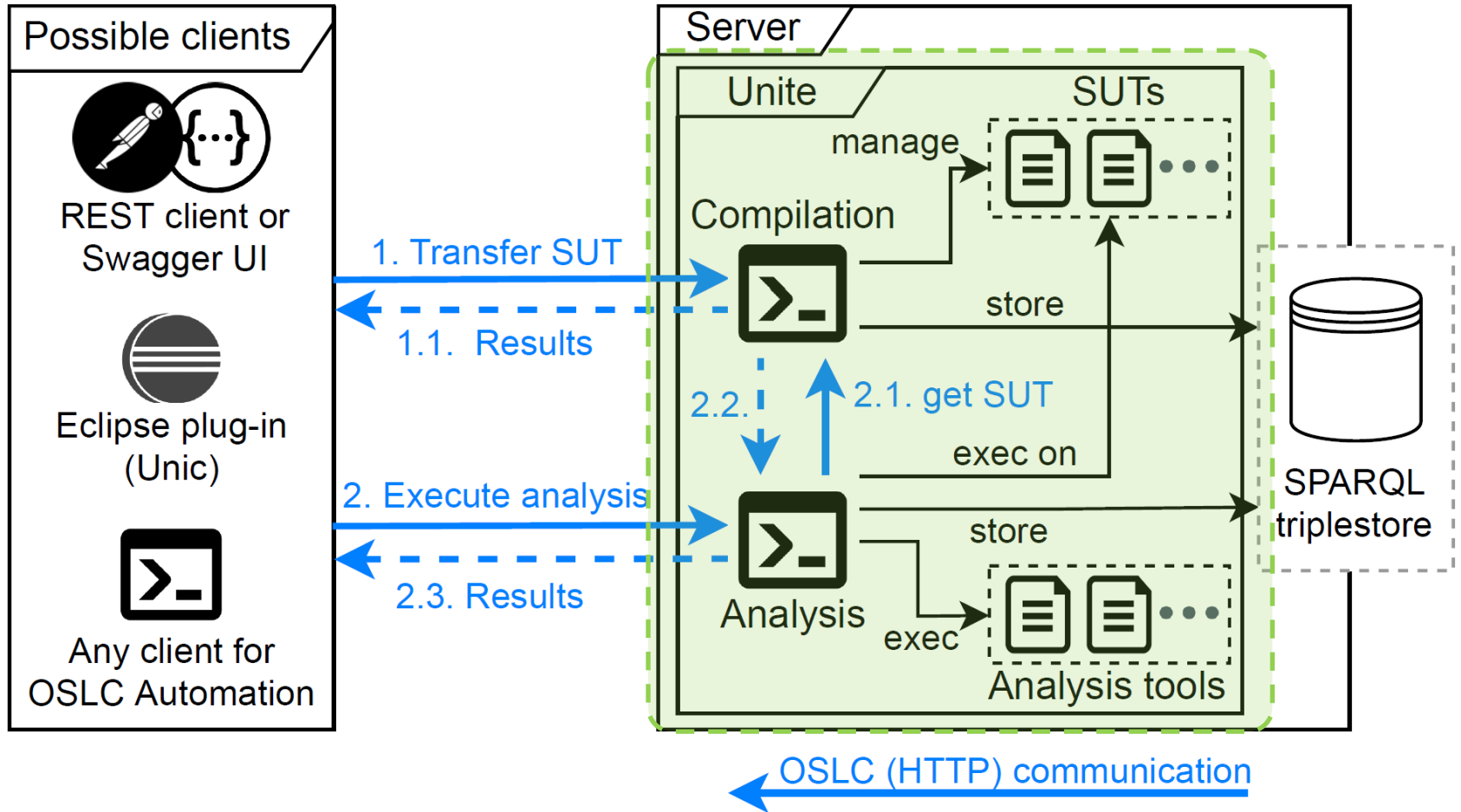
- Outputs of the execution

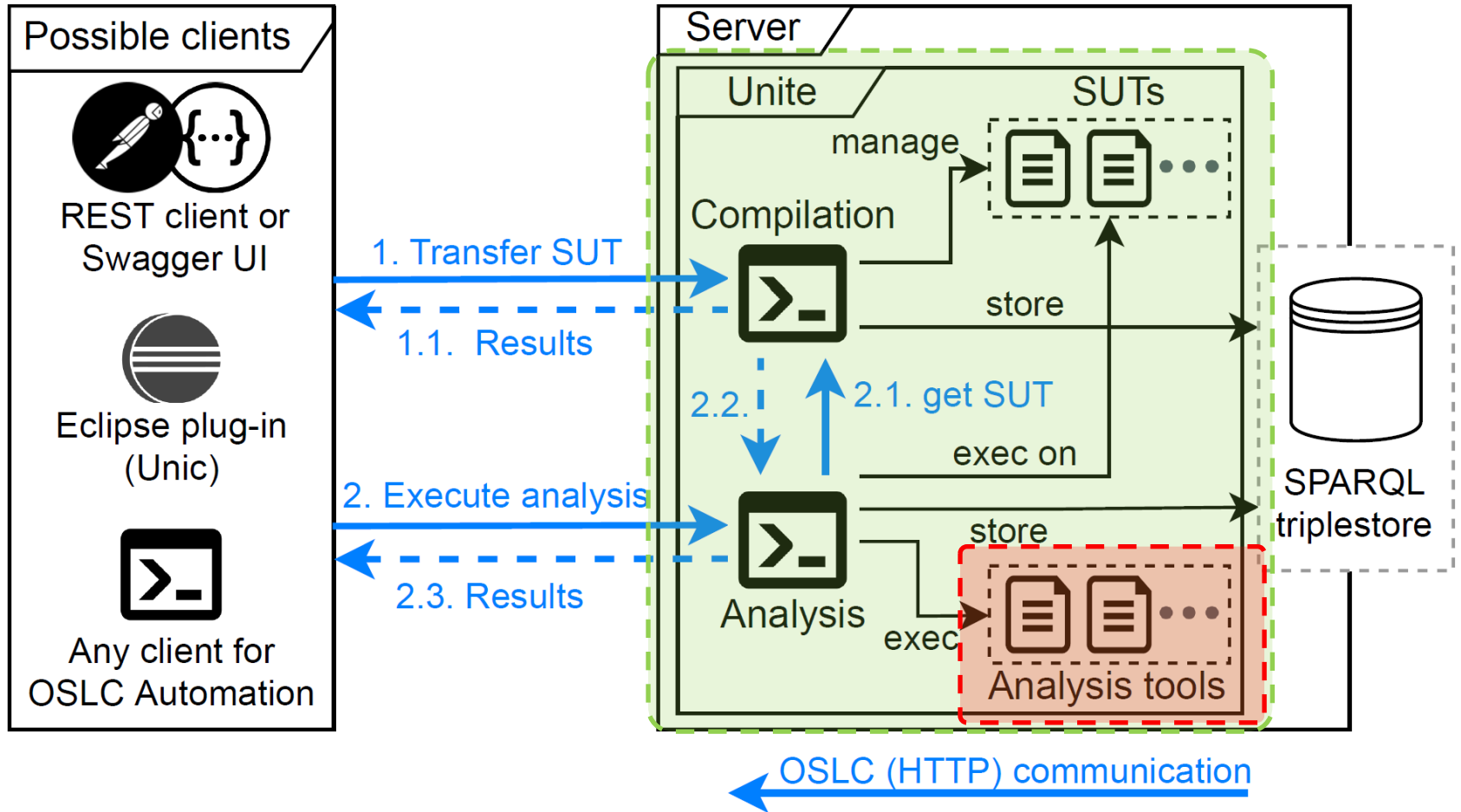


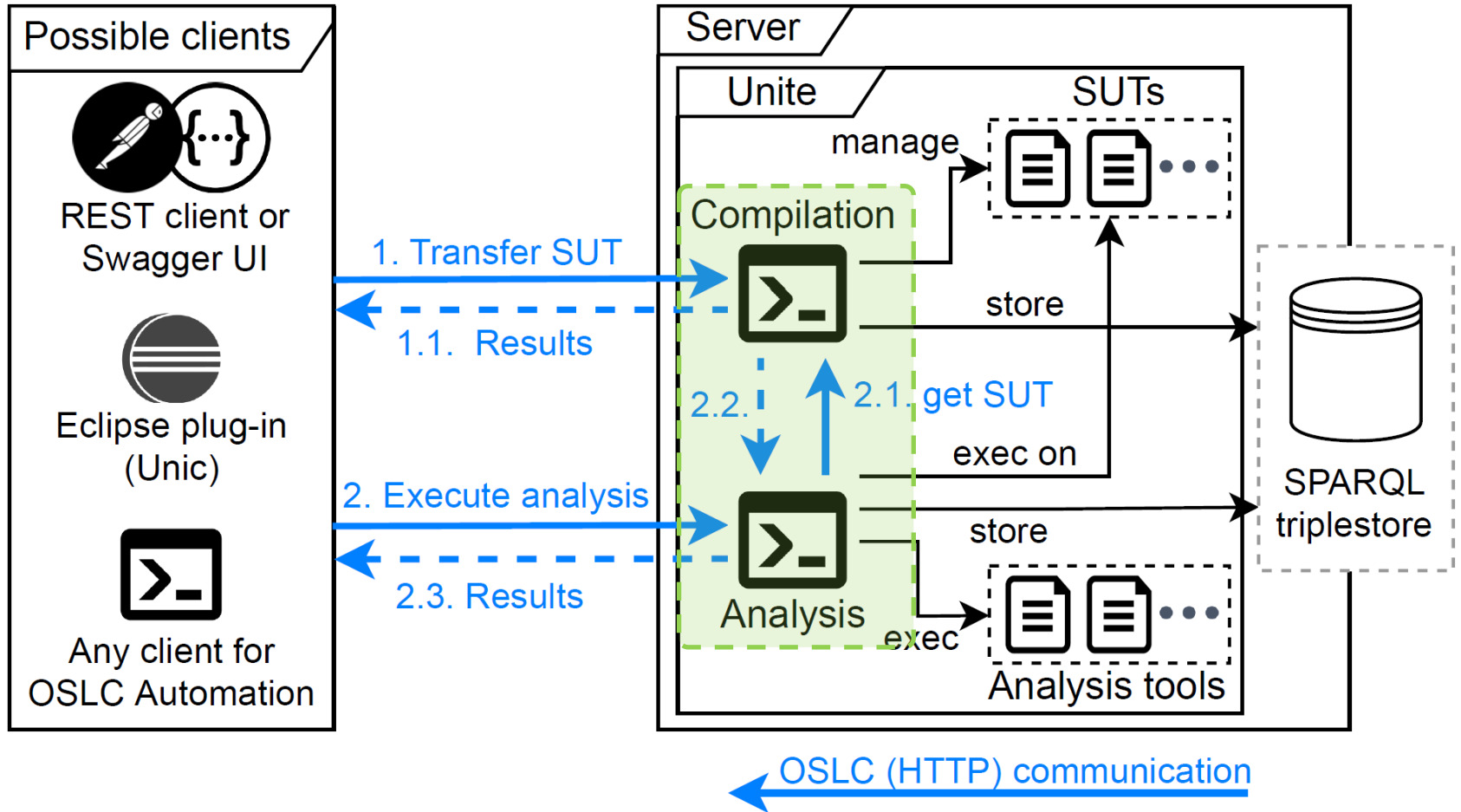
Translating **command-line to OSLC Automation**:

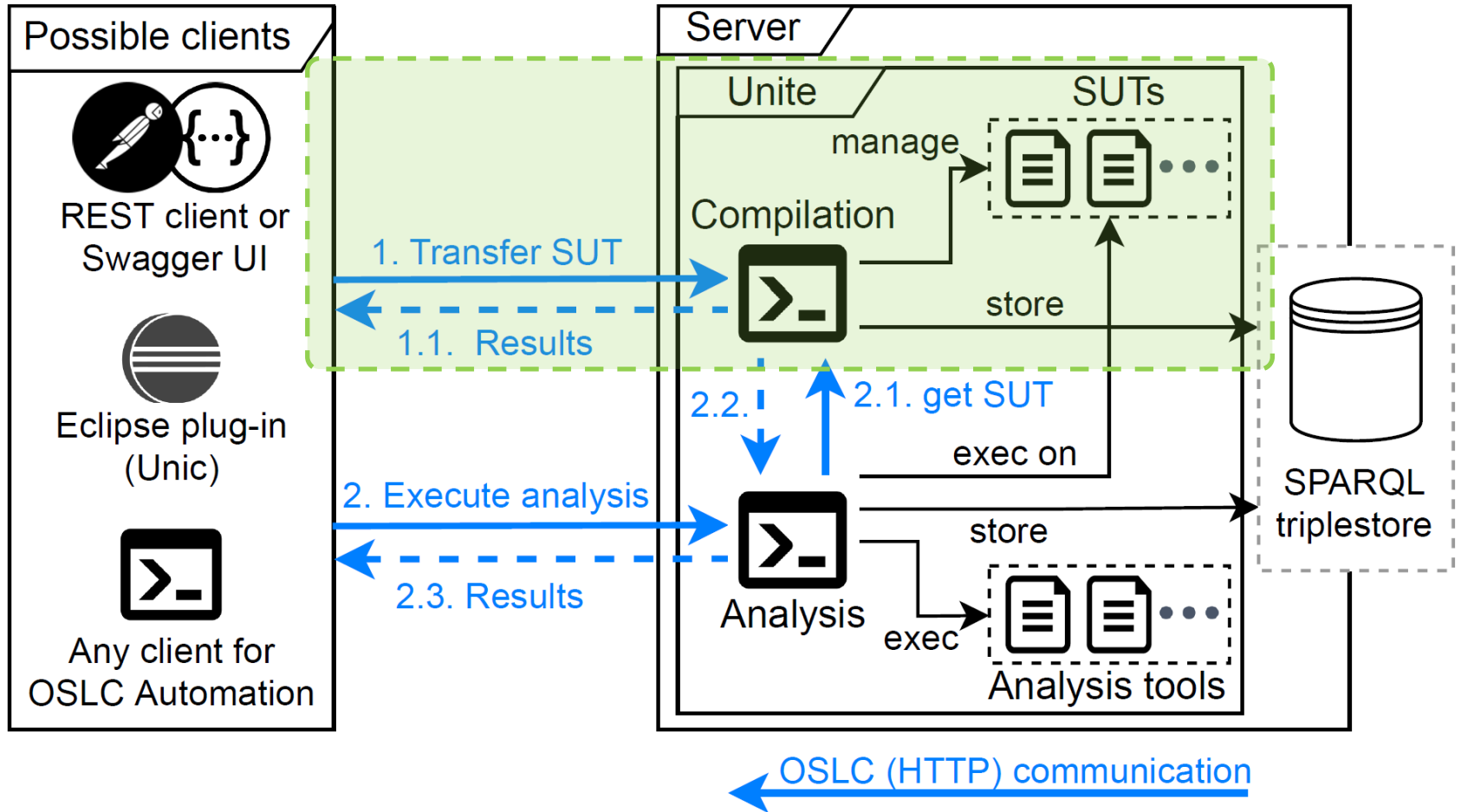


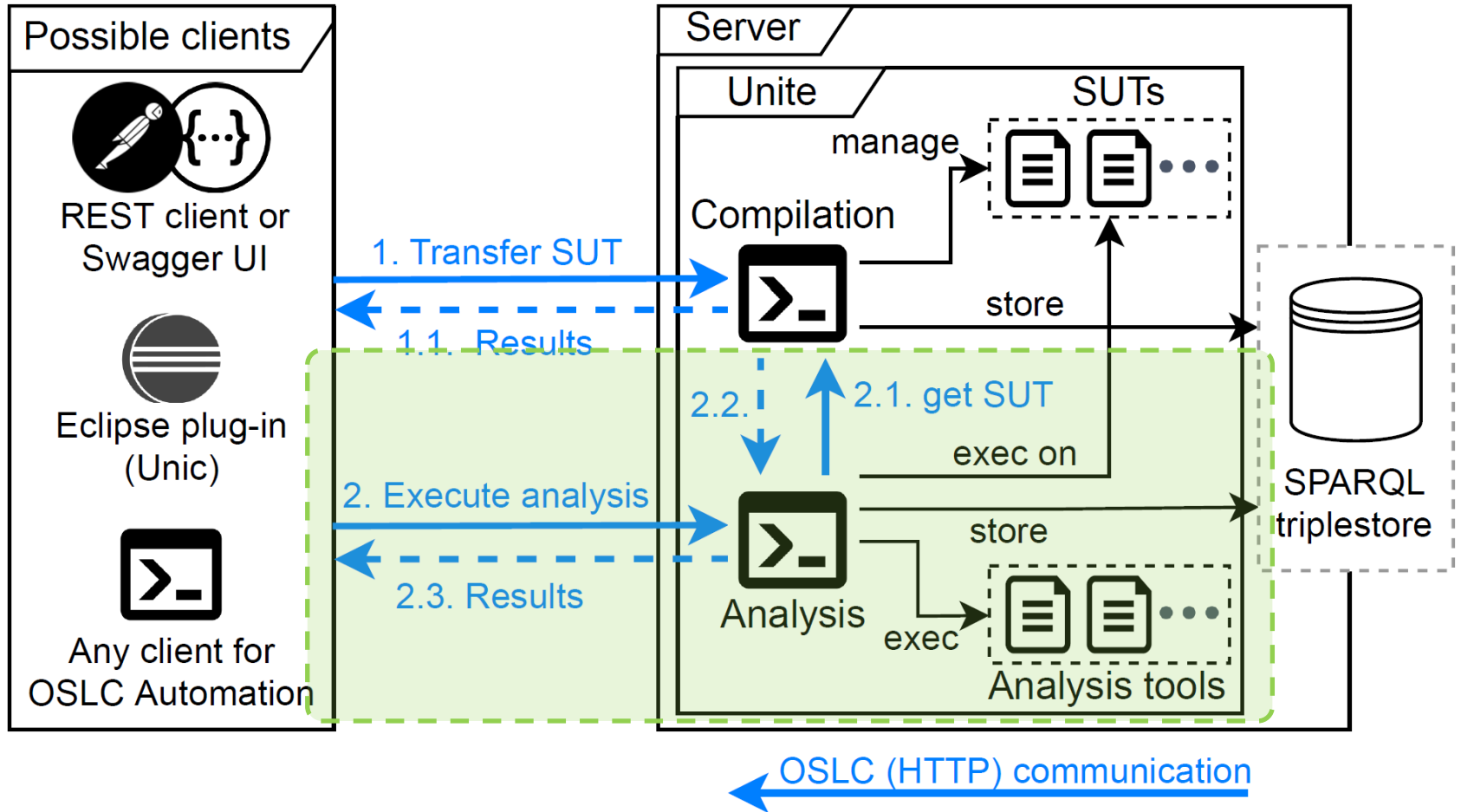


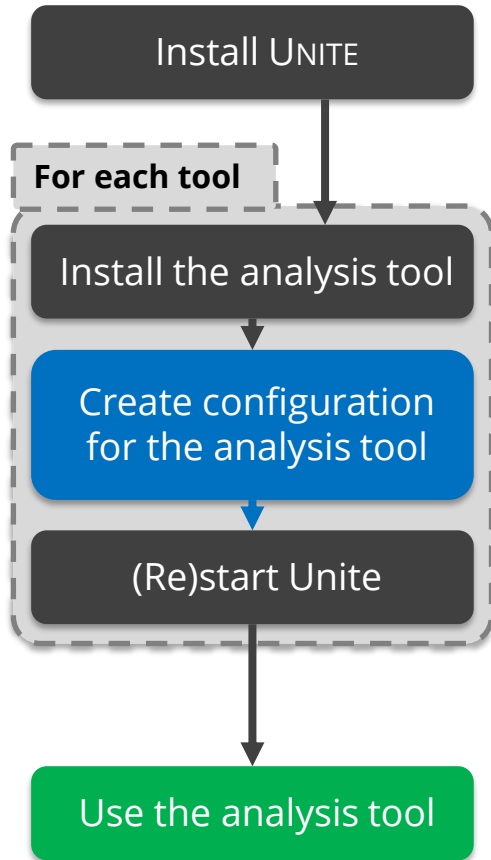


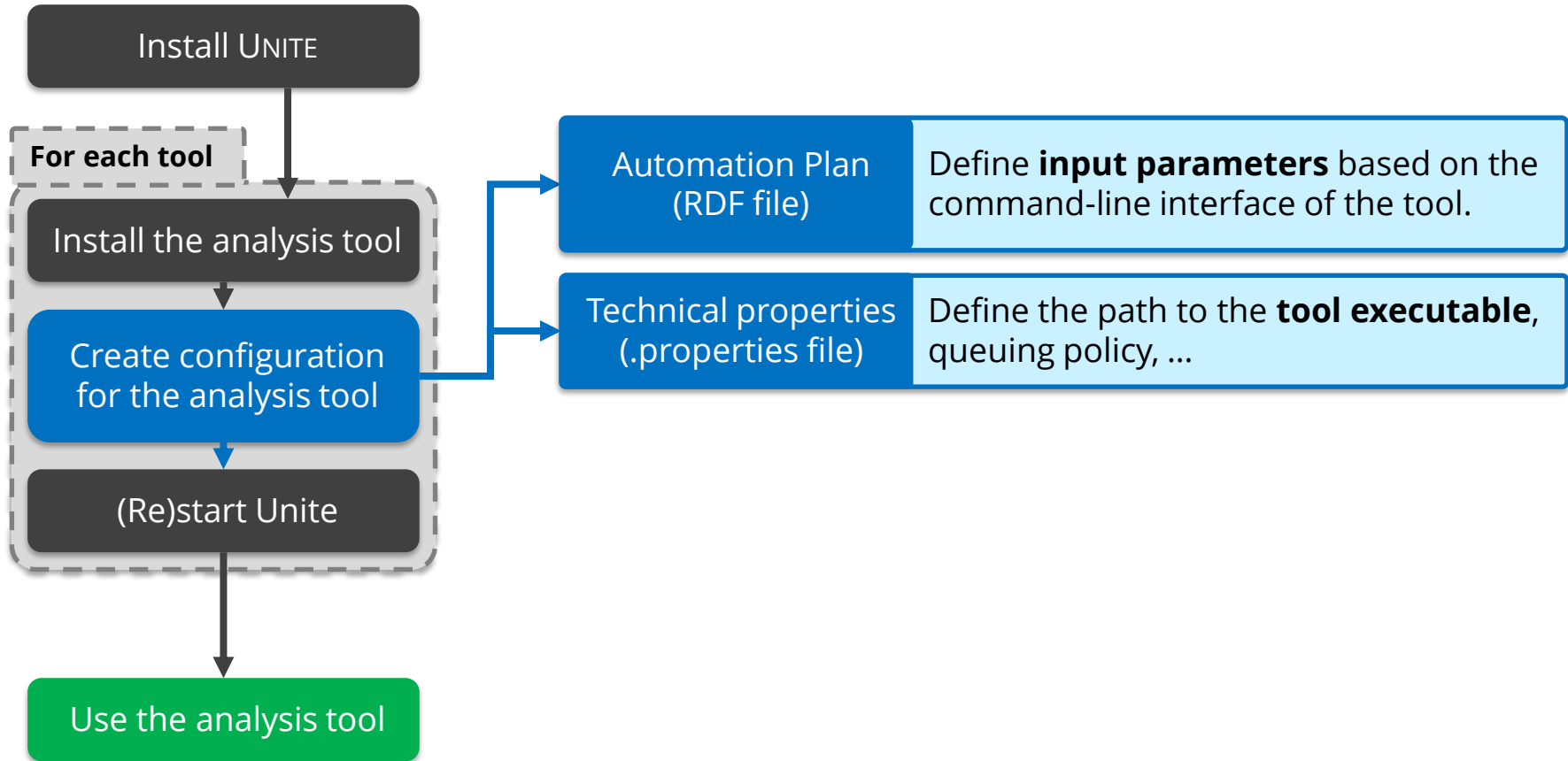


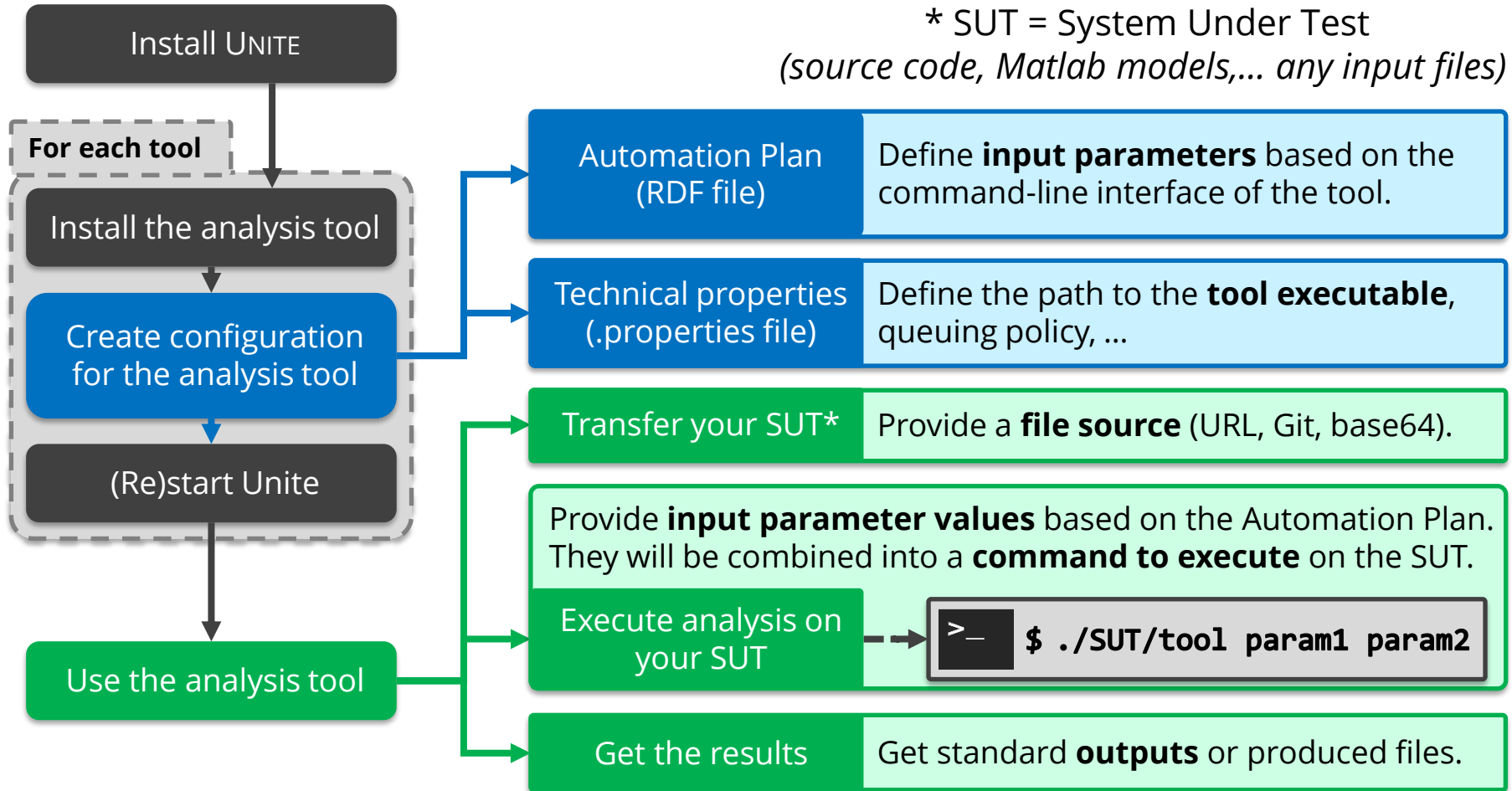


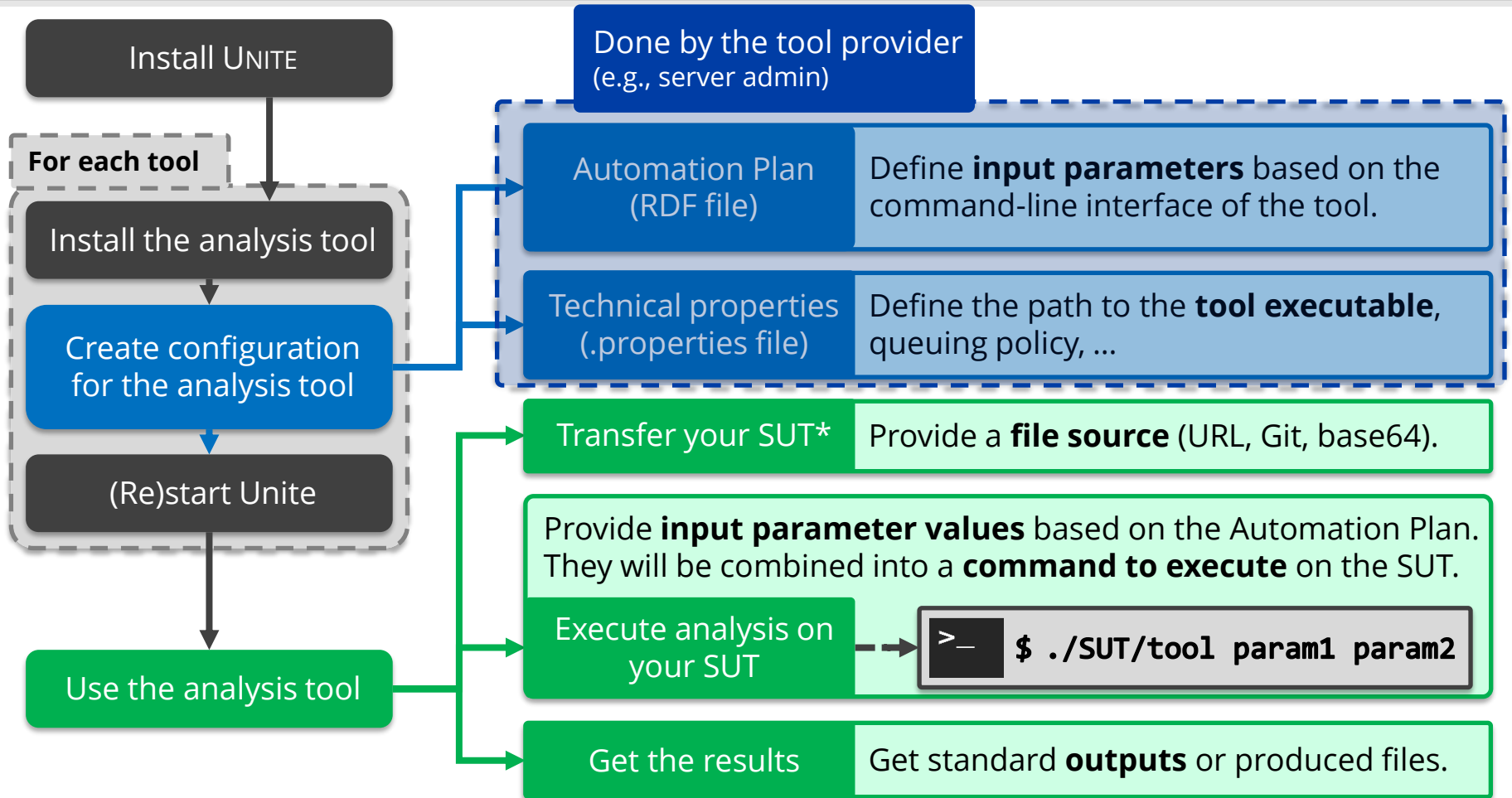


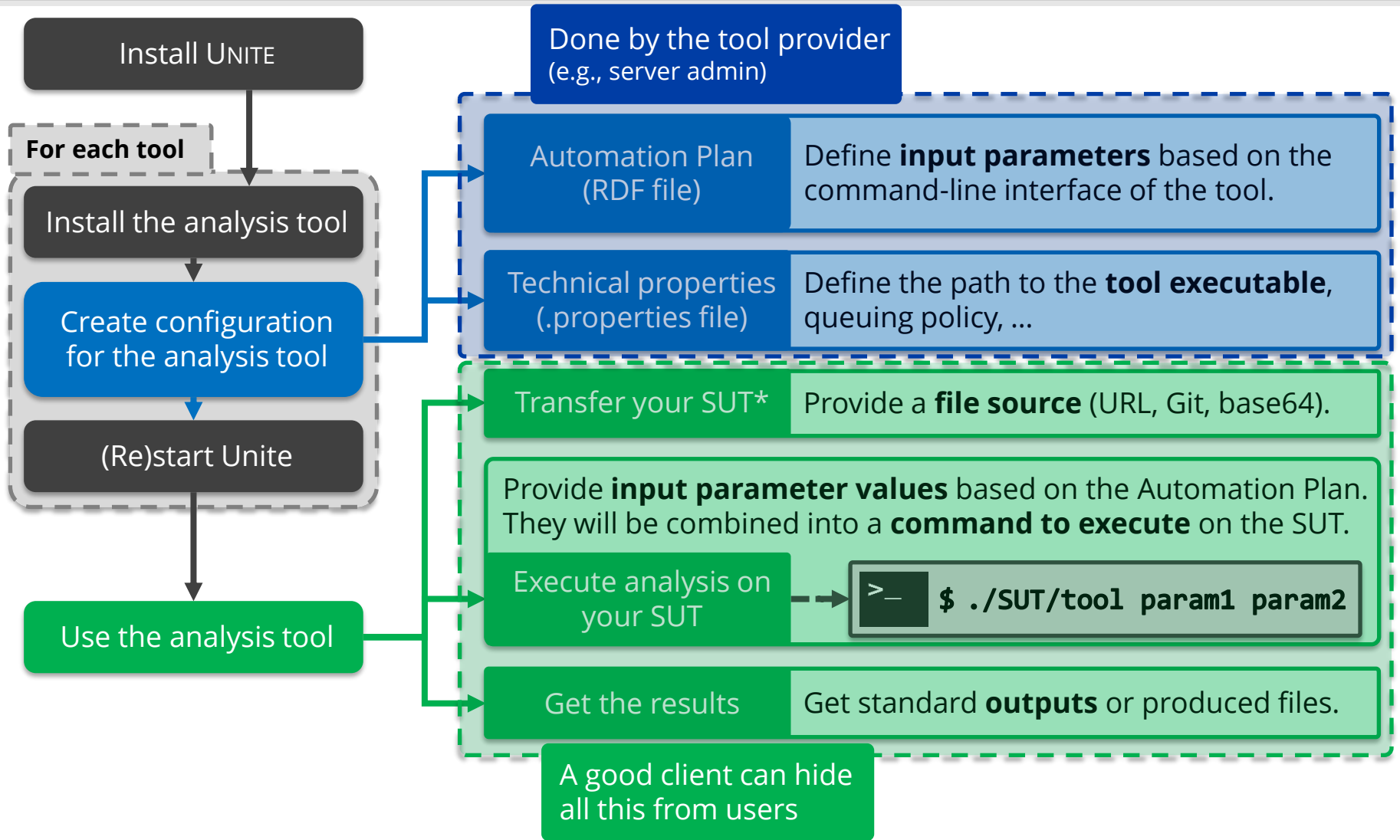












ANaConDA interface:

```
run.sh <analyser> <sut-launch-command> <sut-inputs>
```

```
1 <AutomationPlan>
2 <identifier>anaconda</identifier>
3 <parameterDefinition>
4   <name>analyser</name>
5   <cmdlinePos datatype="#int">1</cmdlinePos>
6   <occurs resource="#Exactly-one"/>
7   <allowedValue>atomrace</allowedValue>
8   <allowedValue>fasttrack</allowedValue>
9 </parameterDefinition><parameterDefinition>
10  <name>SutLaunchCommand</name>
11  <cmdlinePos datatype="#int">2</cmdlinePos>
12  <occurs resource="#Zero-or-One"/>
13  <defaultValue>True</defaultValue>
14 </parameterDefinition><parameterDefinition>
15  <name>SutInputParameters</name>
16  <cmdlinePos datatype="#int">3</cmdlinePos>
17  <occurs resource="#Zero-or-One"/>
18 </parameterDefinition>
19 </AutomationPlan>
```


ANaConDA interface:

```
run.sh <analyser> <sut-launch-command> <sut-inputs>
```

```
1 <AutomationPlan>
2 <identifier>anaconda</identifier>
3 <parameterDefinition>
4   <name>analyser</name>
5   <cmdlinePos datatype="#int">1</cmdlinePos>
6   <occurs resource="#Exactly-one"/>
7   <allowedValue>atomrace</allowedValue>
8   <allowedValue>fasttrack</allowedValue>
9 </parameterDefinition><parameterDefinition>
10  <name>SutLaunchCommand</name>
11  <cmdlinePos datatype="#int">2</cmdlinePos>
12  <occurs resource="#Zero-or-One"/>
13  <defaultValue>True</defaultValue>
14 </parameterDefinition><parameterDefinition>
15  <name>SutInputParameters</name>
16  <cmdlinePos datatype="#int">3</cmdlinePos>
17  <occurs resource="#Zero-or-One"/>
18 </parameterDefinition>
19 </AutomationPlan>
```

ANaConDA interface:

```
run.sh <analyser> <sut-launch-command> <sut-inputs>
```

```
1 <AutomationPlan>
2 <identifier>anaconda</identifier>
3 <parameterDefinition>
4   <name>analyser</name>
5   <cmdlinePos datatype="#int">1</cmdlinePos>
6   <occurs resource="#Exactly-one"/>
7   <allowedValue>atomrace</allowedValue>
8   <allowedValue>fasttrack</allowedValue>
9 </parameterDefinition><parameterDefinition>
10  <name>SutLaunchCommand</name>
11  <cmdlinePos datatype="#int">2</cmdlinePos>
12  <occurs resource="#Zero-or-One"/>
13  <defaultValue>True</defaultValue>
14 </parameterDefinition><parameterDefinition>
15  <name>SutInputParameters</name>
16  <cmdlinePos datatype="#int">3</cmdlinePos>
17  <occurs resource="#Zero-or-One"/>
18 </parameterDefinition>
19 </AutomationPlan>
```

- Successfully tested with a range of tools
 - FACEBOOK INFER, VALGRIND, ANACONDA, PERUN, GREP, THETA, HILITE, SYMBIOTIC,...

- **Successfully tested** with a range of tools
 - FACEBOOK INFER, VALGRIND, ANACONDA, PERUN, GREP, THETA, HILITE, SYMBIOTIC,...
- **Deployed** in Honeywell
 - Used for an in-house tool **HILITE** (a test vector generation tool)
 - Multiple use cases
 - 1) Brno**
 - 2) India**
 - 3) China**

- **Successfully tested** with a range of tools
 - FACEBOOK INFER, VALGRIND, ANACONDA, PERUN, GREP, THETA, HiLITE, SYMBIOTIC,...
- **Deployed** in Honeywell
 - Used for an in-house tool **HiLITE** (a test vector generation tool)
 - Multiple use cases
 - 1) Brno**
 - Used 2 clients: **Web UI**, **FORREQ** (analysis aggregation tool)
 - **18 new users** and ~8hours of time saved per month
 - 2) India**
 - 3) China**

- **Successfully tested** with a range of tools
 - FACEBOOK INFER, VALGRIND, ANACONDA, PERUN, GREP, THETA, HiLiTE, SYMBIOTIC,...
- **Deployed** in Honeywell
 - Used for an in-house tool **HiLiTE** (a test vector generation tool)
 - Multiple use cases
 - 1) Brno**
 - Used 2 clients: **Web UI, FORREQ** (analysis aggregation tool)
 - **18 new users** and ~8hours of time saved per month
 - 2) India**
 - Used a **custom PowerShell client** for CI/CD pipelines
 - Enabled use of **HiLiTE in CI/CD** (incompatibilities, licensing, ...)
 - 3) China**

- **Successfully tested** with a range of tools
 - FACEBOOK INFER, VALGRIND, ANACONDA, PERUN, GREP, THETA, HiLiTE, SYMBIOTIC,...
- **Deployed** in Honeywell
 - Used for an in-house tool **HiLiTE** (a test vector generation tool)
 - Multiple use cases
 - 1) Brno**
 - Used 2 clients: **Web UI**, **FORREQ** (analysis aggregation tool)
 - **18 new users** and ~8hours of time saved per month
 - 2) India**
 - Used a **custom PowerShell client** for CI/CD pipelines
 - Enabled use of **HiLiTE in CI/CD** (incompatibilities, licensing, ...)
 - 3) China**
 - Used **UNIC** in the Eclipse IDE
 - Allowed direct **access to HiLiTE** functionality (export control)

Thank you for your attention!

&

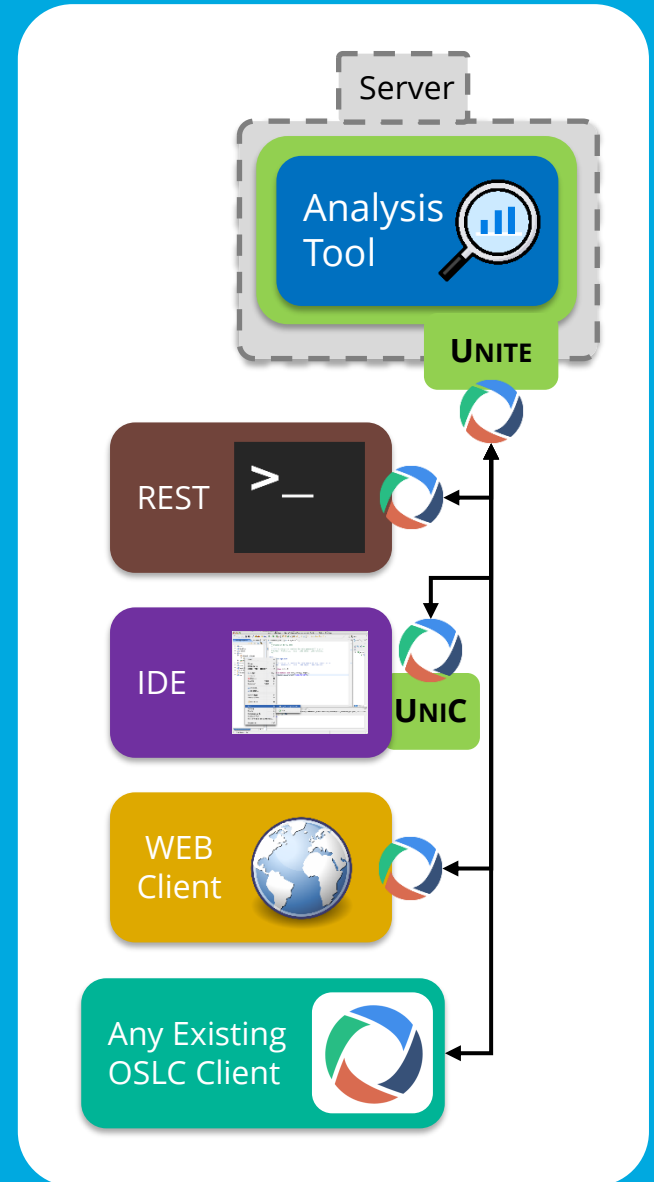
See you in a coffee break

Speaker: Ondřej Vašíček (ivasicek@fit.vutbr.cz)

Paper: <https://doi.org/10.1145/3540250.3558939>

UNITE repo: <https://pajda.fit.vutbr.cz/verifit/unite>

Demo VM: <https://doi.org/10.5281/zenodo.6074820>

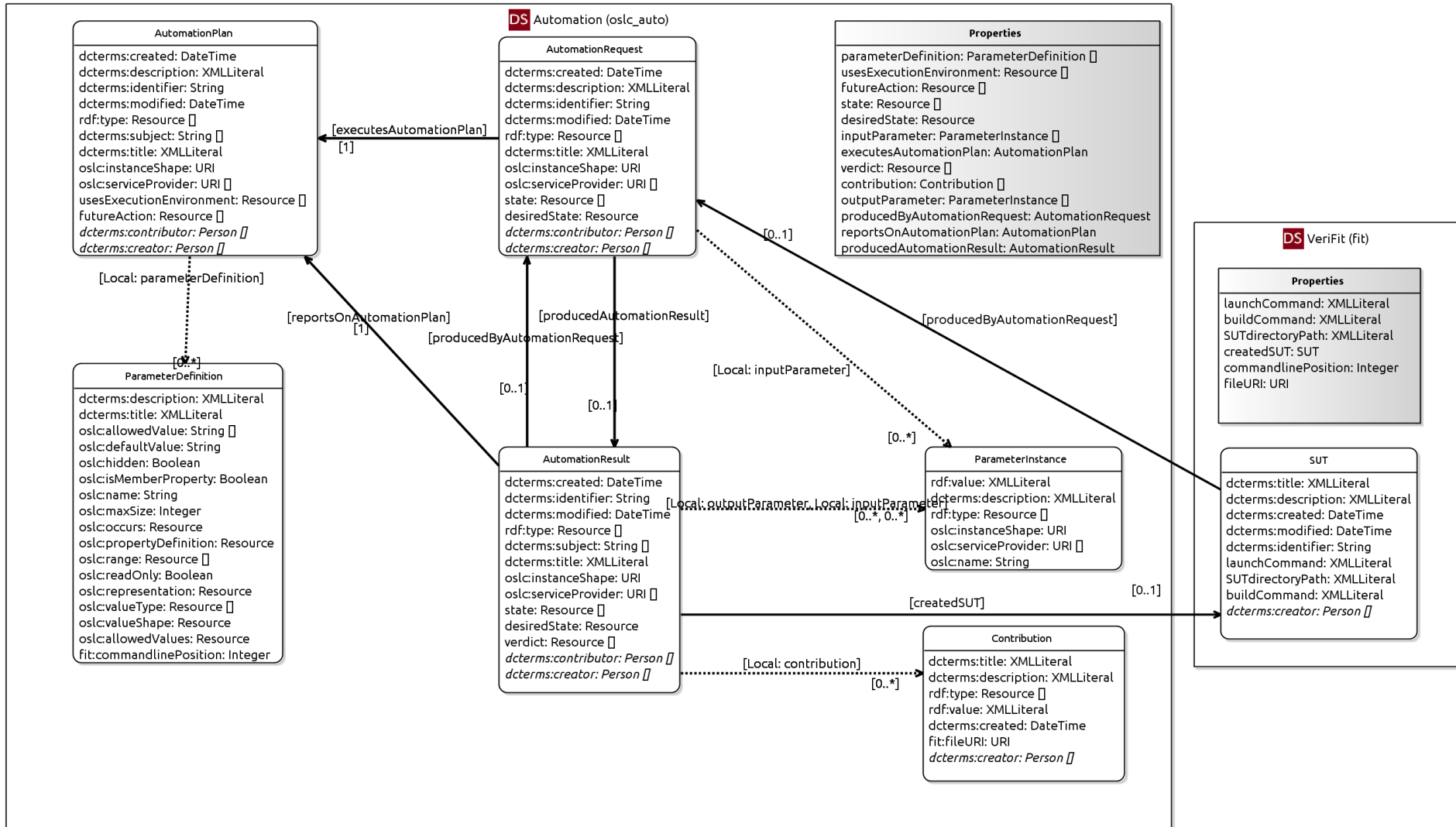


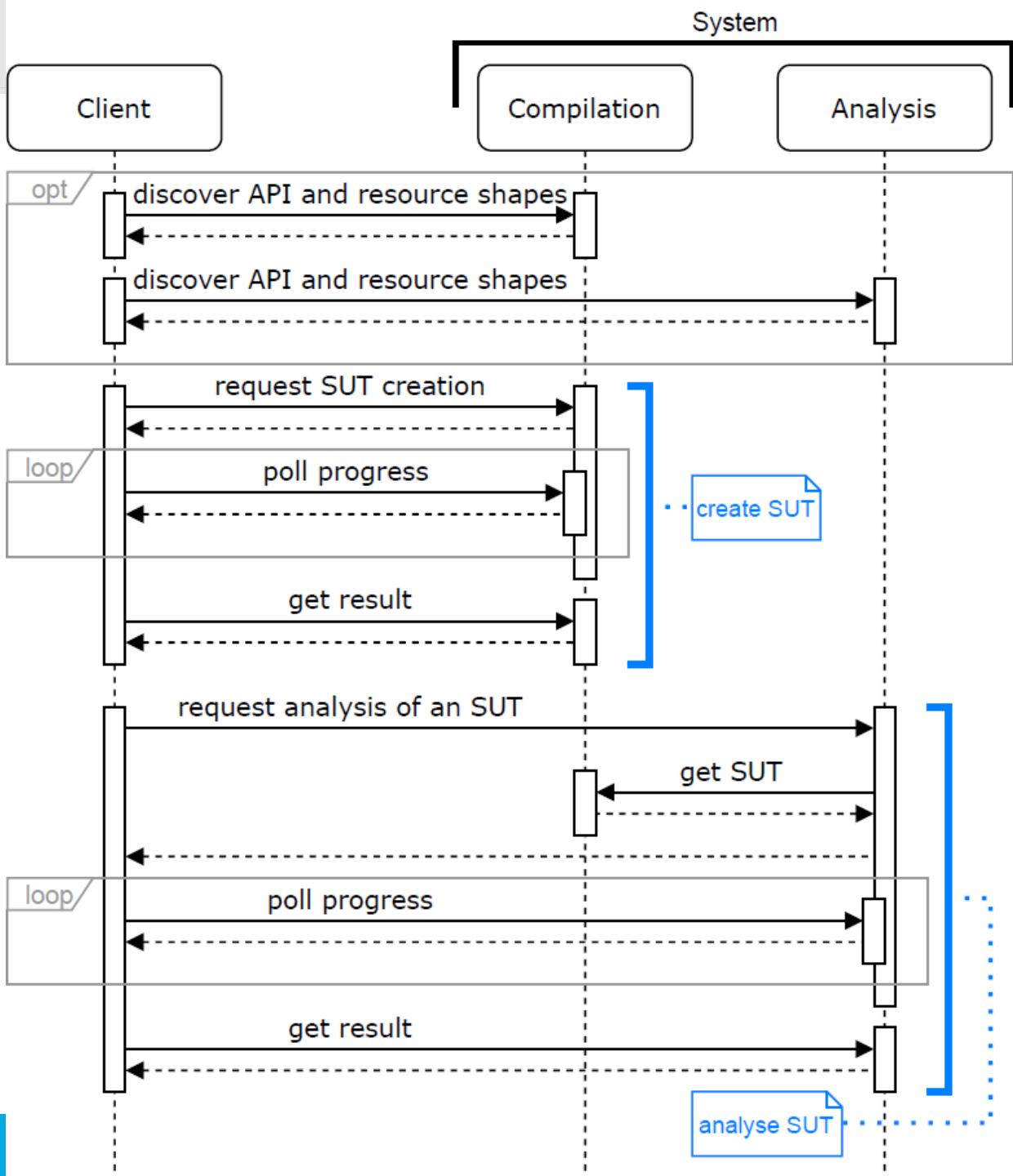
As executed by Unite:

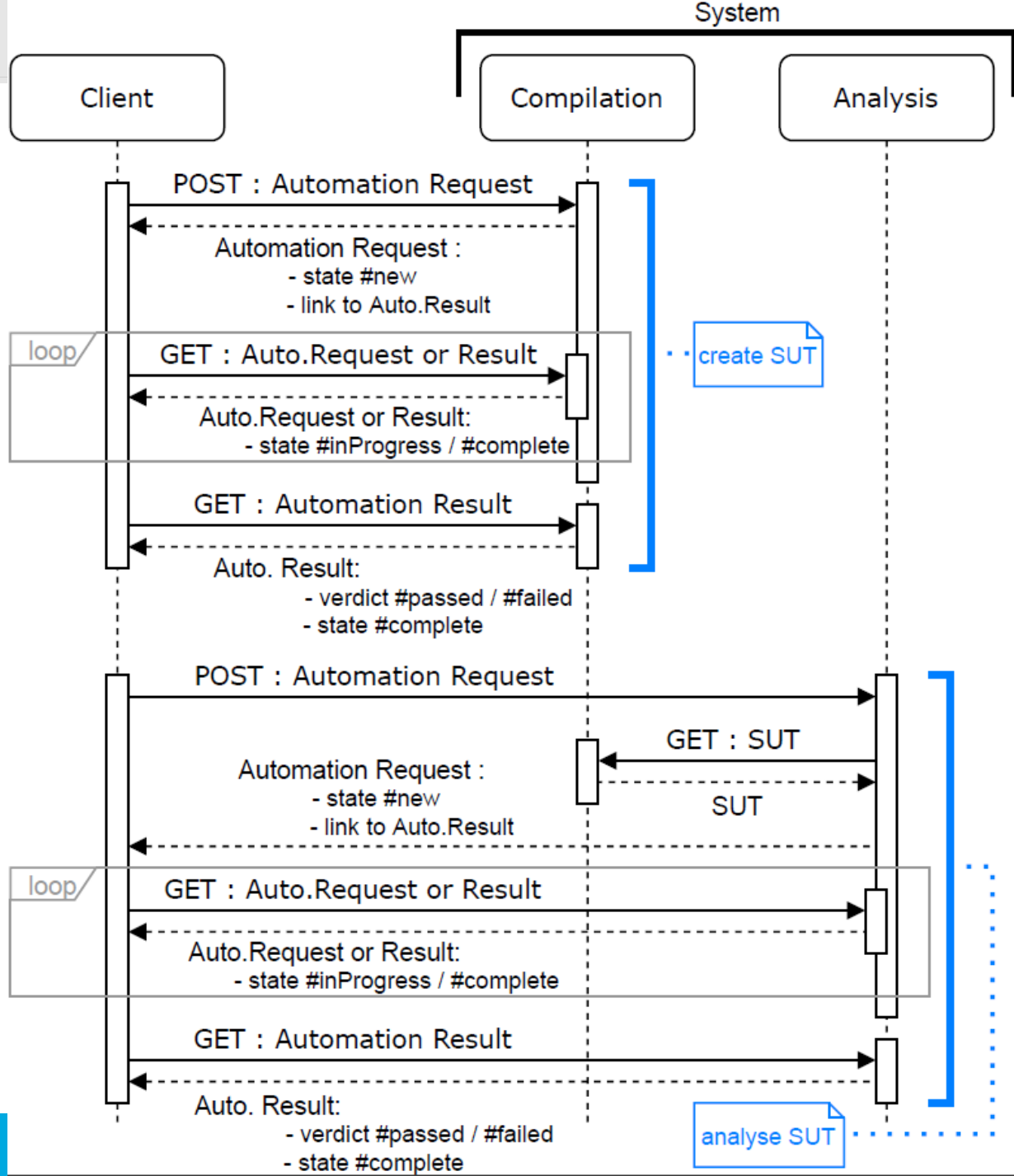
```
anaconda/run.sh atomrace ./hello "Hello World!"
```

```
1 <AutomationRequest><!-- POST TO ../analysis/ctrAutoRequest -->
2 <execAutomationPlan res="../AutomationPlans/anaconda"/>
3 <inputParameter>
4   <name>analyser</name><value>atomrace</value>
5 </inputParameter><inputParameter>
6   <name>SUT</name><value>../URI/of/the/analysed/SUT</value>
7 </inputParameter><inputParameter>
8   <name>SutInputParameters</name><value>"Hello World!"</value>
9 </inputParameter>
10 </AutomationRequest>
```

```
1 <AutomationRequest><!--POST TO ../compilation/ctrAutoRequest-->
2 <execAutomationPlan res="../AutomationPlans/SutCreation"/>
3 <inputParameter>
4   <name>sourceUrl</name><value>http://../sut.zip</value>
5 </inputParameter><inputParameter>
6   <name>unpackZip</name><value>>true</value>
7 </inputParameter><inputParameter>
8   <name>buildCommand</name><value>make</value>
9 </inputParameter><inputParameter>
10  <name>launchCommand</name><value>./hello</value>
11 </inputParameter>
12 </AutomationRequest>
```







create SUT

analyse SUT