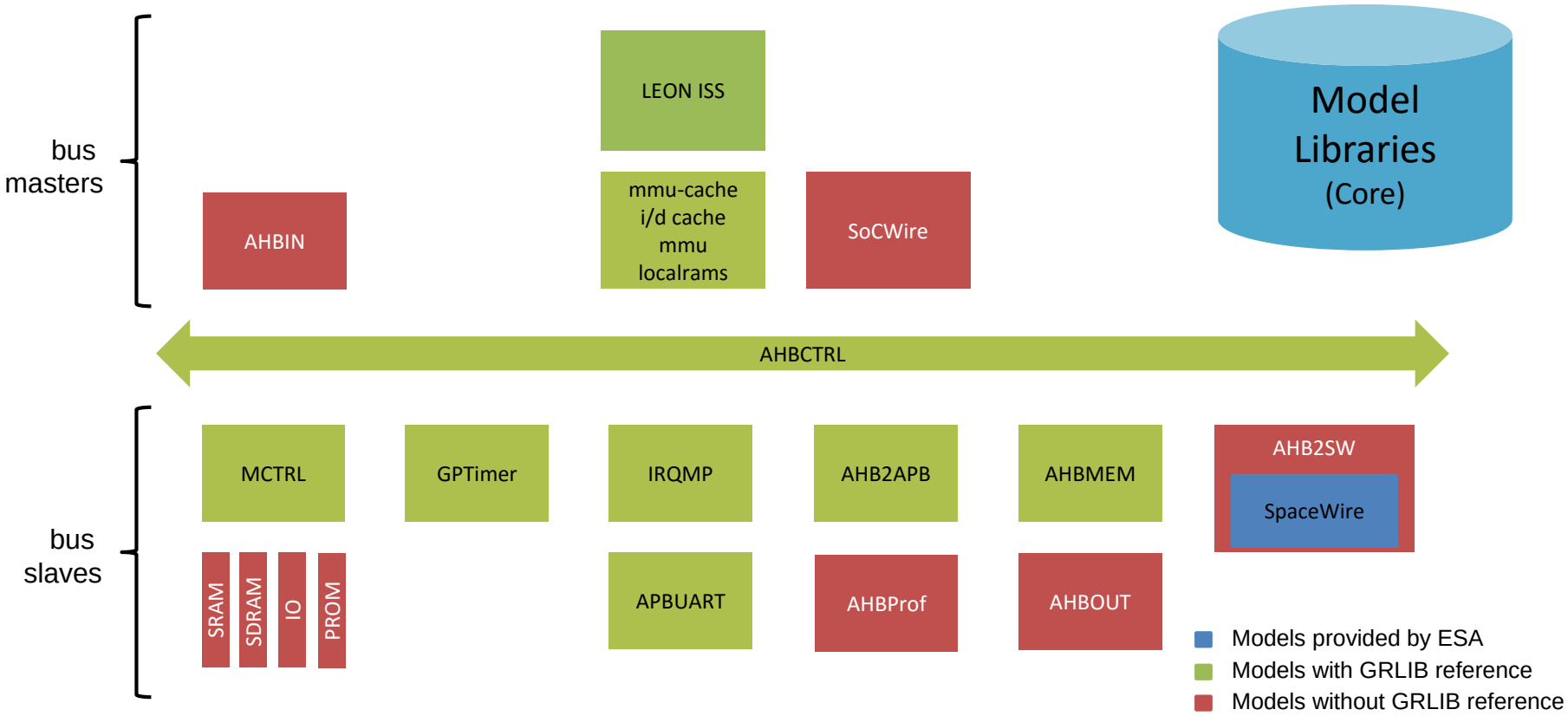
Universal Scripting Interface for SystemC

Rolf Meyer, E.I.S., TU Braunschweig

SoCRocket TLM Models

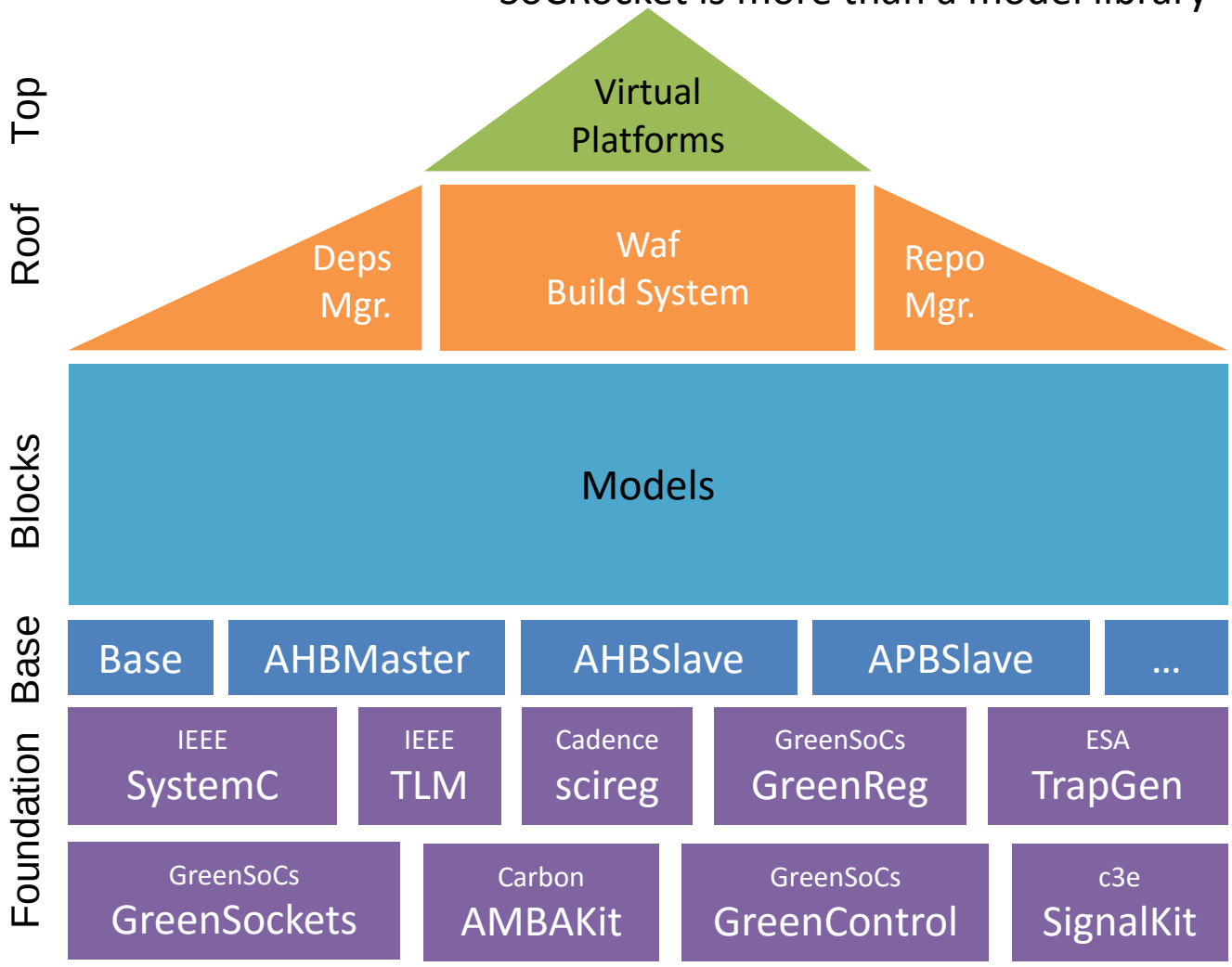
All models developed with RTL equivalents as blueprint

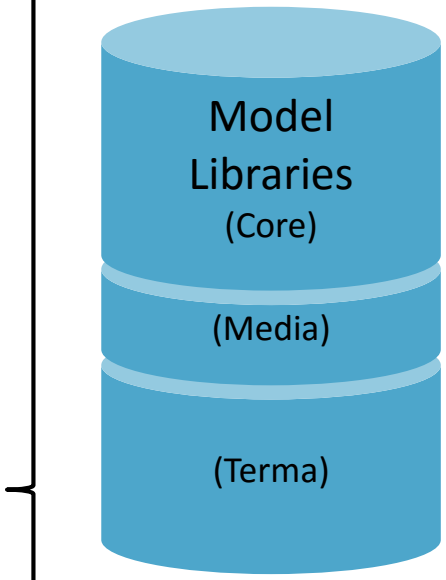


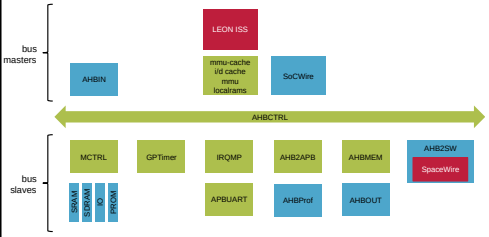
- Models available in loosely timed (LT), and approximately timed (AT) flavor of TLM2.0.
- ESA Reference TLM Platform

SoCRocket - The building blocks

SoCRocket is more than a model library







Desired Features

- Automating procedures
- Configuring simulation parameters
- Access runtime simulation information
- Easy testing integration
- Interactive introspection

Desired Features

- Automating procedures
- Configuring simulation parameters
- Access runtime simulation information
- Easy testing integration
- Interactive introspection

→ **Perfect tasks for scripting languages**

Scripting Language

- Defining variables/option values
- Recording/executing command sequences
- Capturing output results
- Branching and looping
- Importing and exporting options and
- Bridging the gaps to other abstraction levels

Available simulators

- Accellera SystemC
- ALDEC Riviera-PRO → TCL
- Cadence ncsim → TCL
- Mentor Graphics QuestaSim → TCL
- Synopsys Platform Architect → TCL
- …

- The choice depends on the problem/programmer

Scripting in EDA/SystemC

Most common: TCL

- Command oriented (Shell like)
- No native OOP
- Therefore not the best user experience

Scripting support in Accellera SystemC

- No integrated language
- Different approaches available

Scripting languages for SystemC

Existing implementations

- TCL
- SystemLua → focused on configuration
- GreenScript → focused on abstract modeling

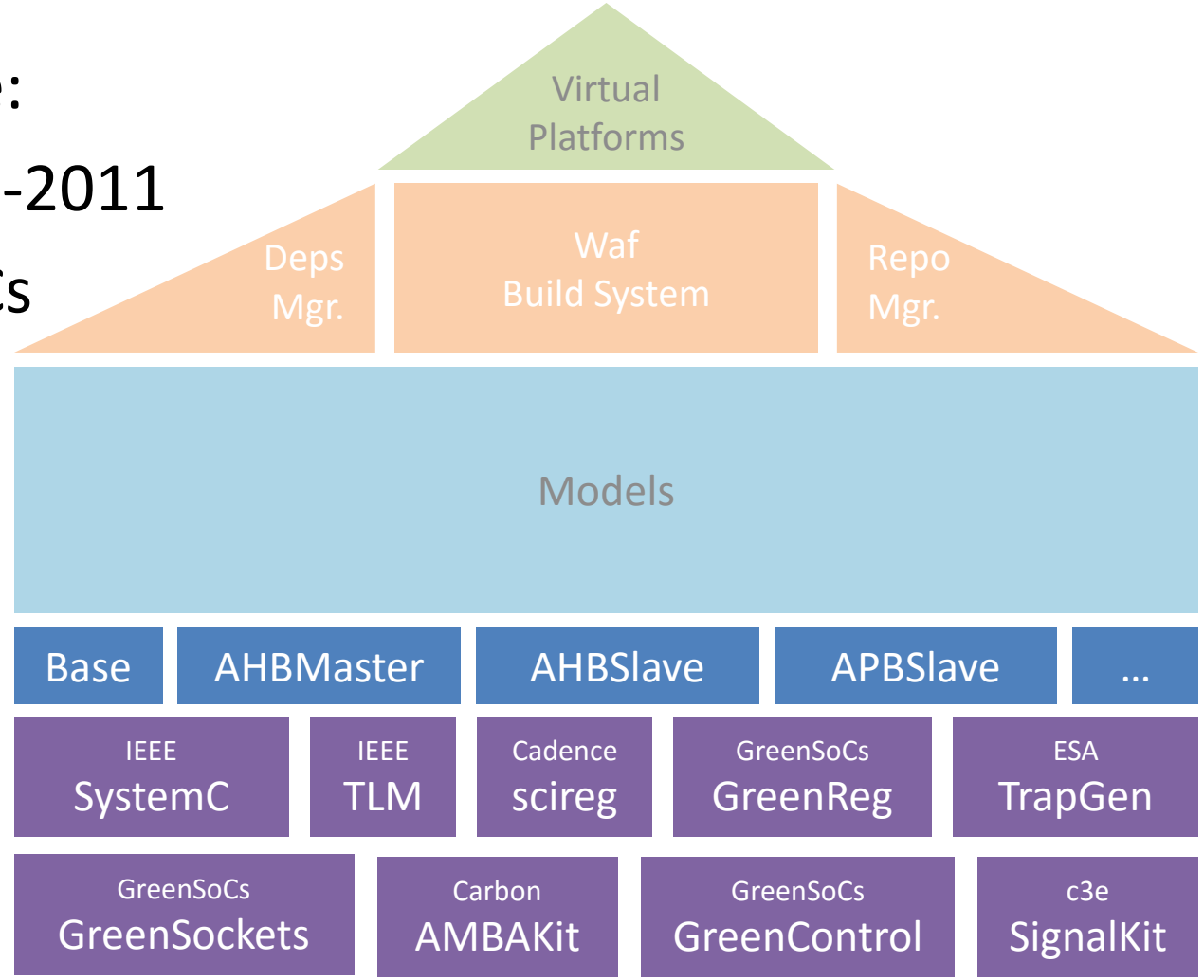
Desired implementation

- depends on the problem/programmer
- we propose Python for its usability

Common SystemC APIs

For example:

- IEEE 1666-2011
- GreenSoCs
- Cadence
- Carbon
- Own
- …



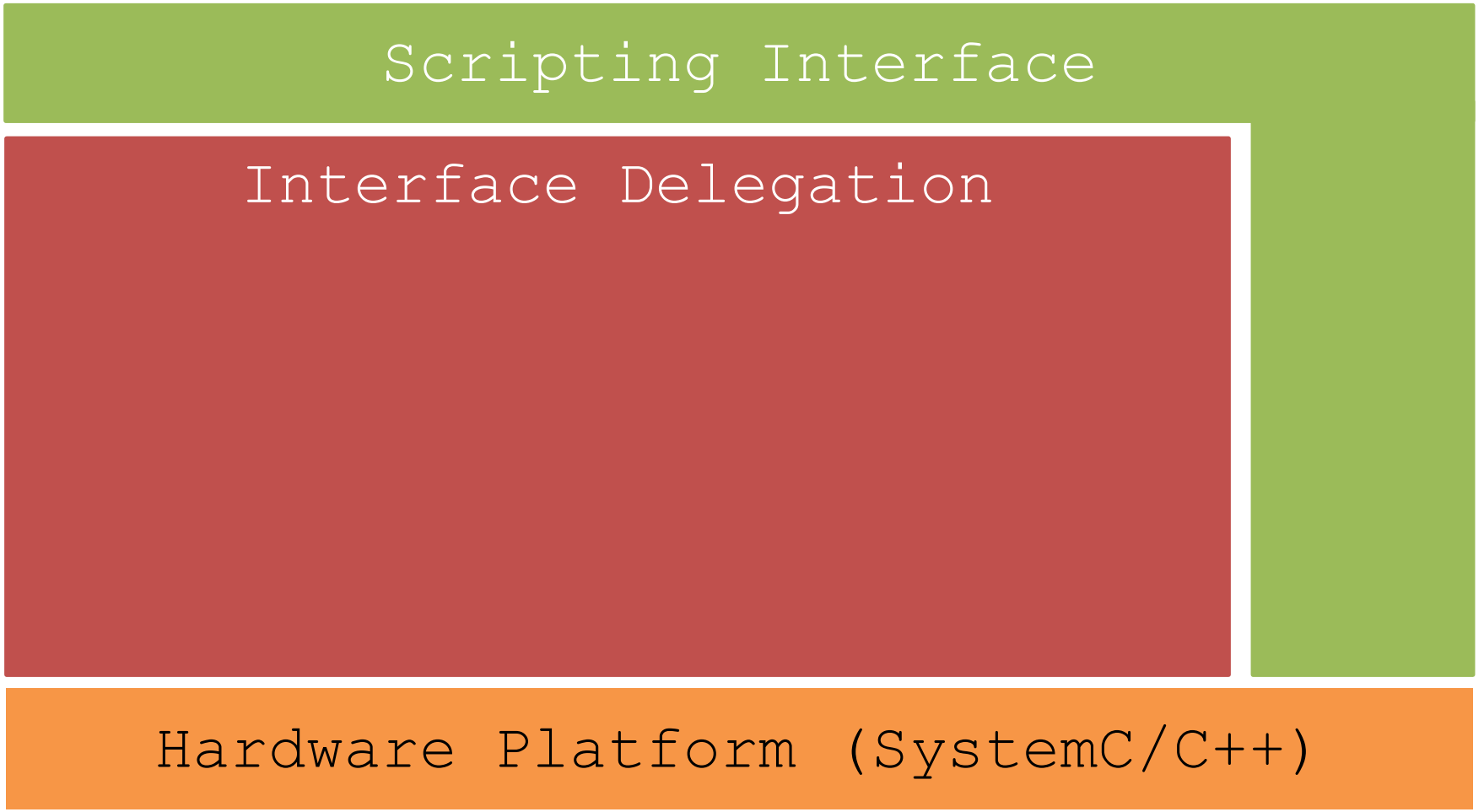
Additional requirements

- Available in multiple Simulators
- Language independent
- Same/similar APIs as in SystemC/C++

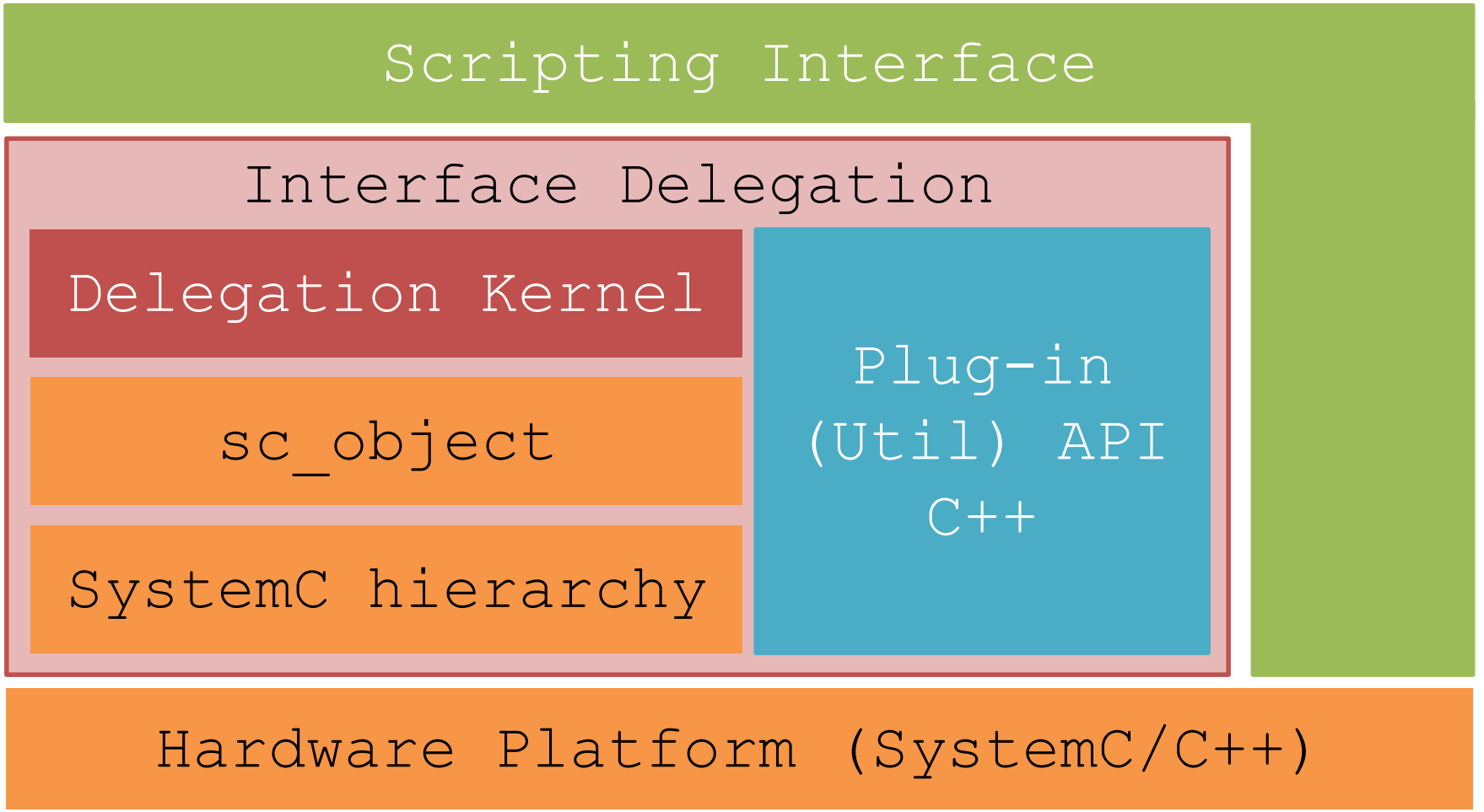
- Addressable via hierarchical module name

```
top.obj.reg
```

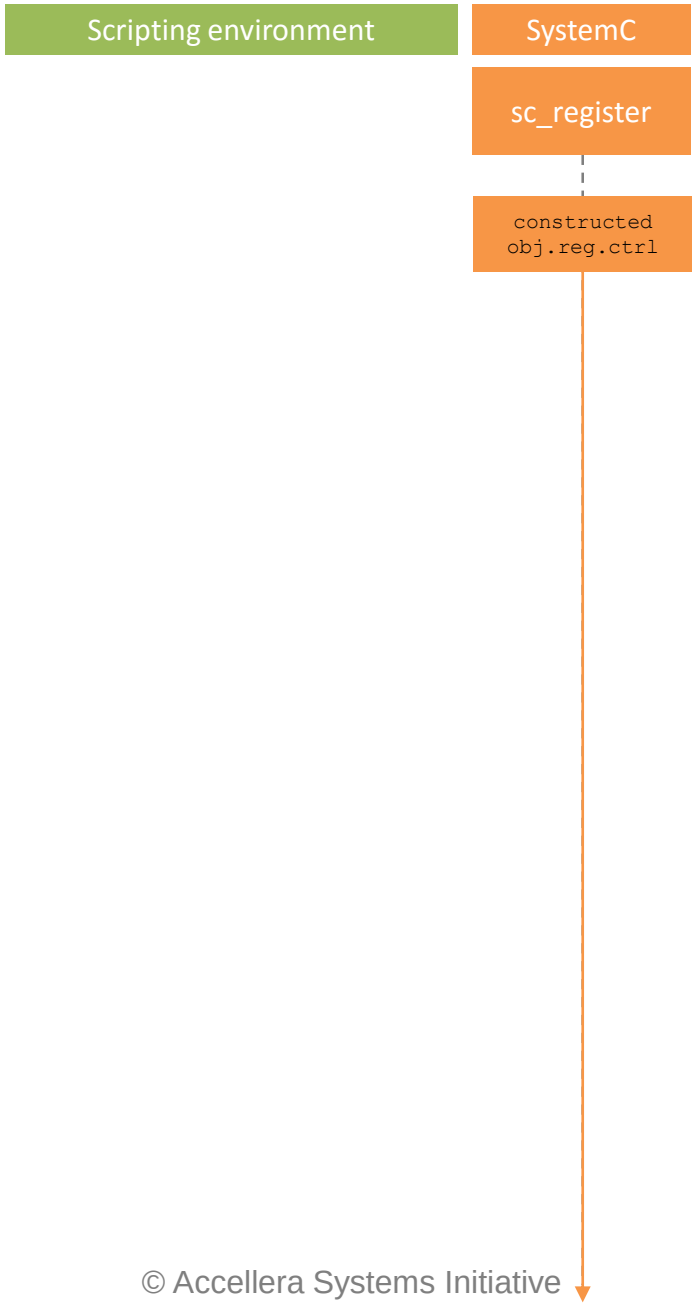
Environment

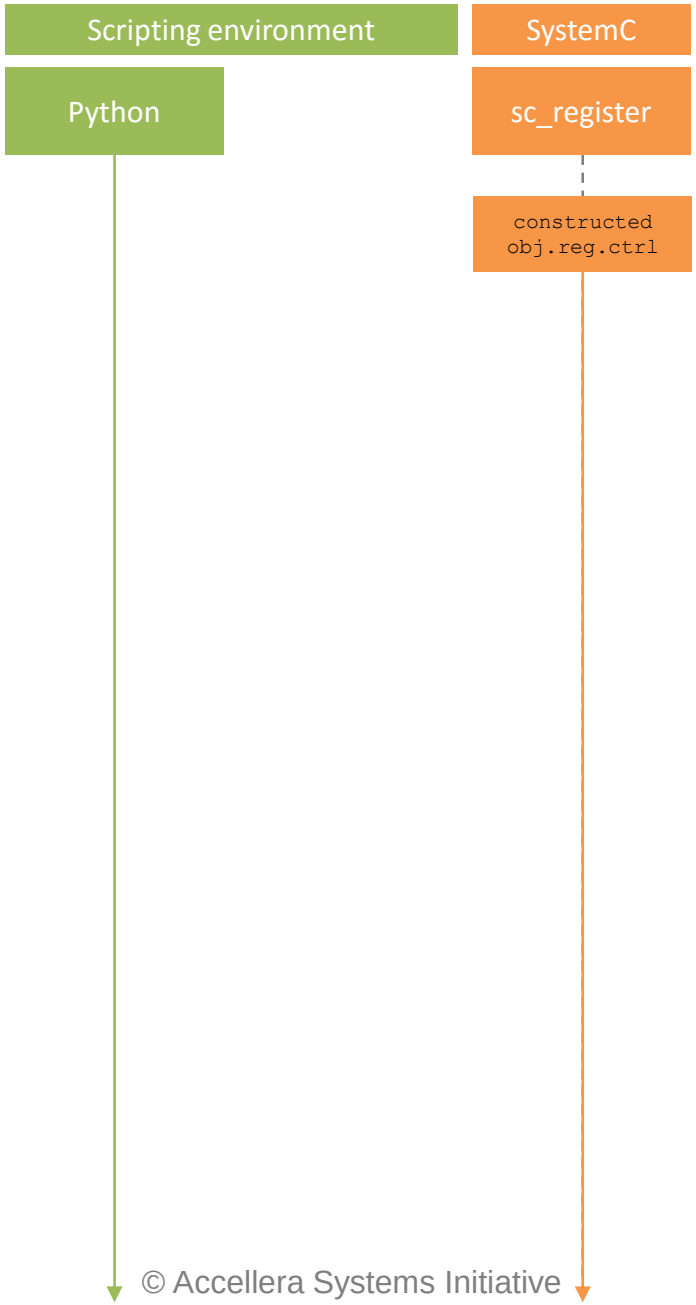


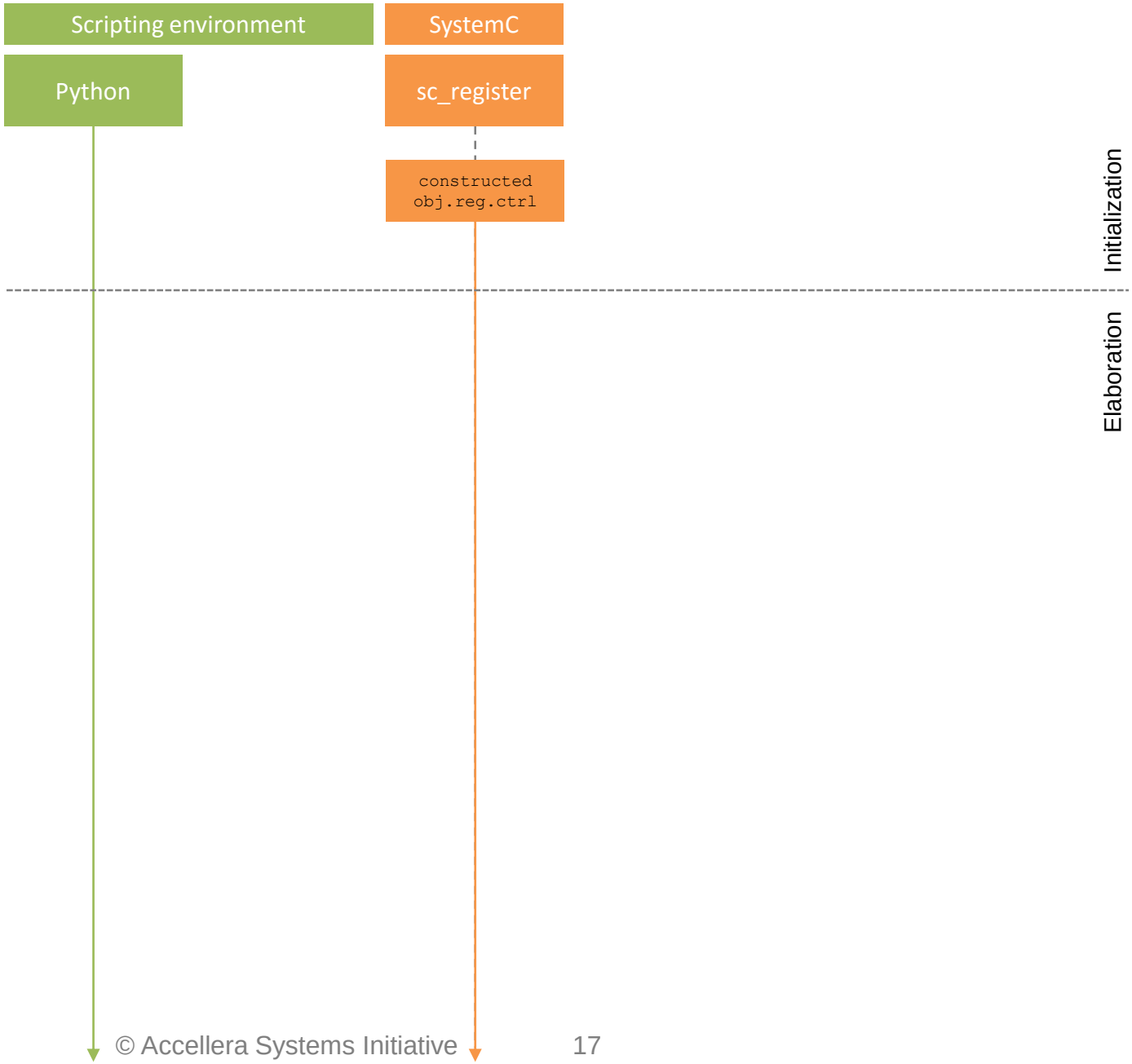
Environment

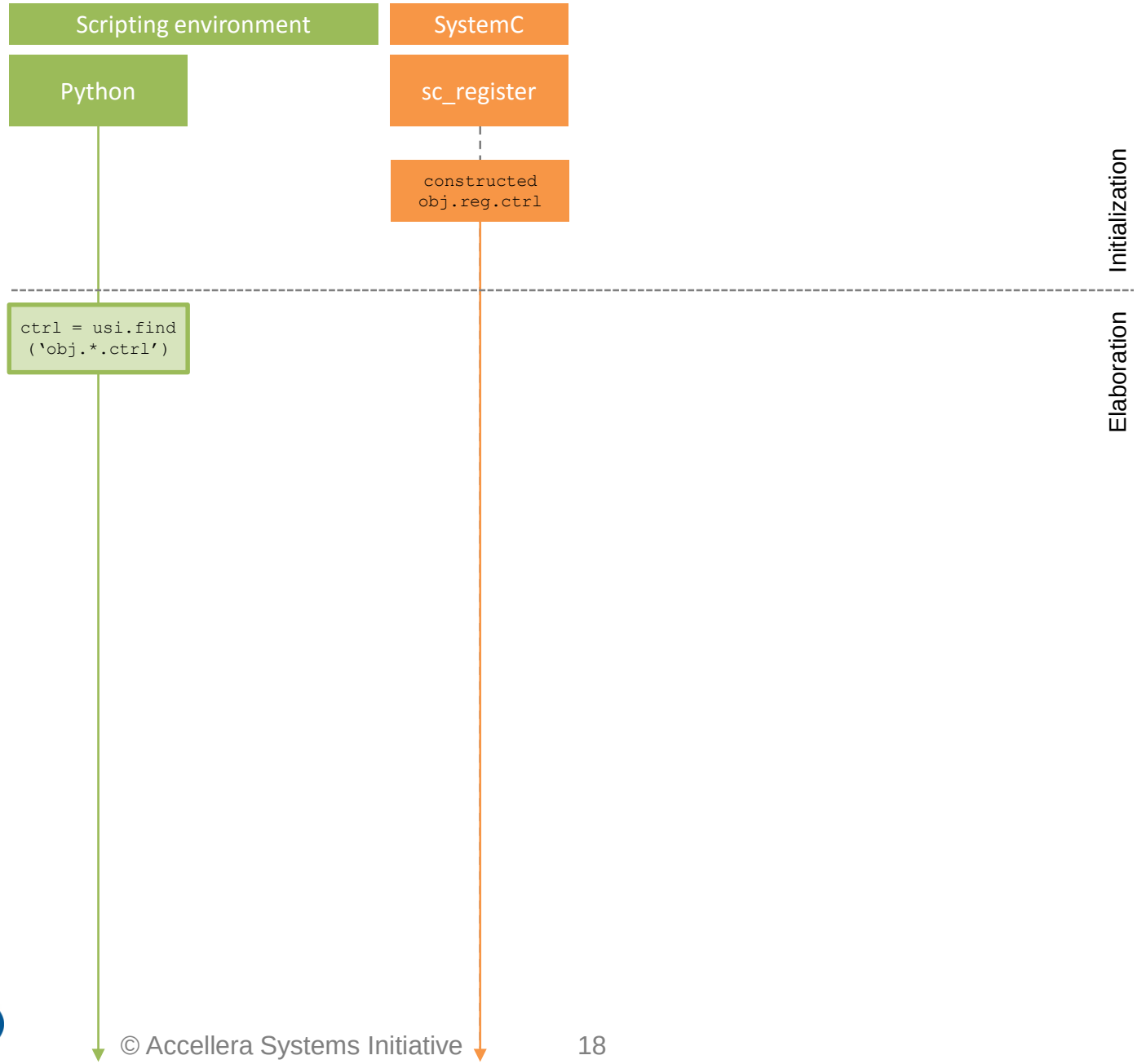


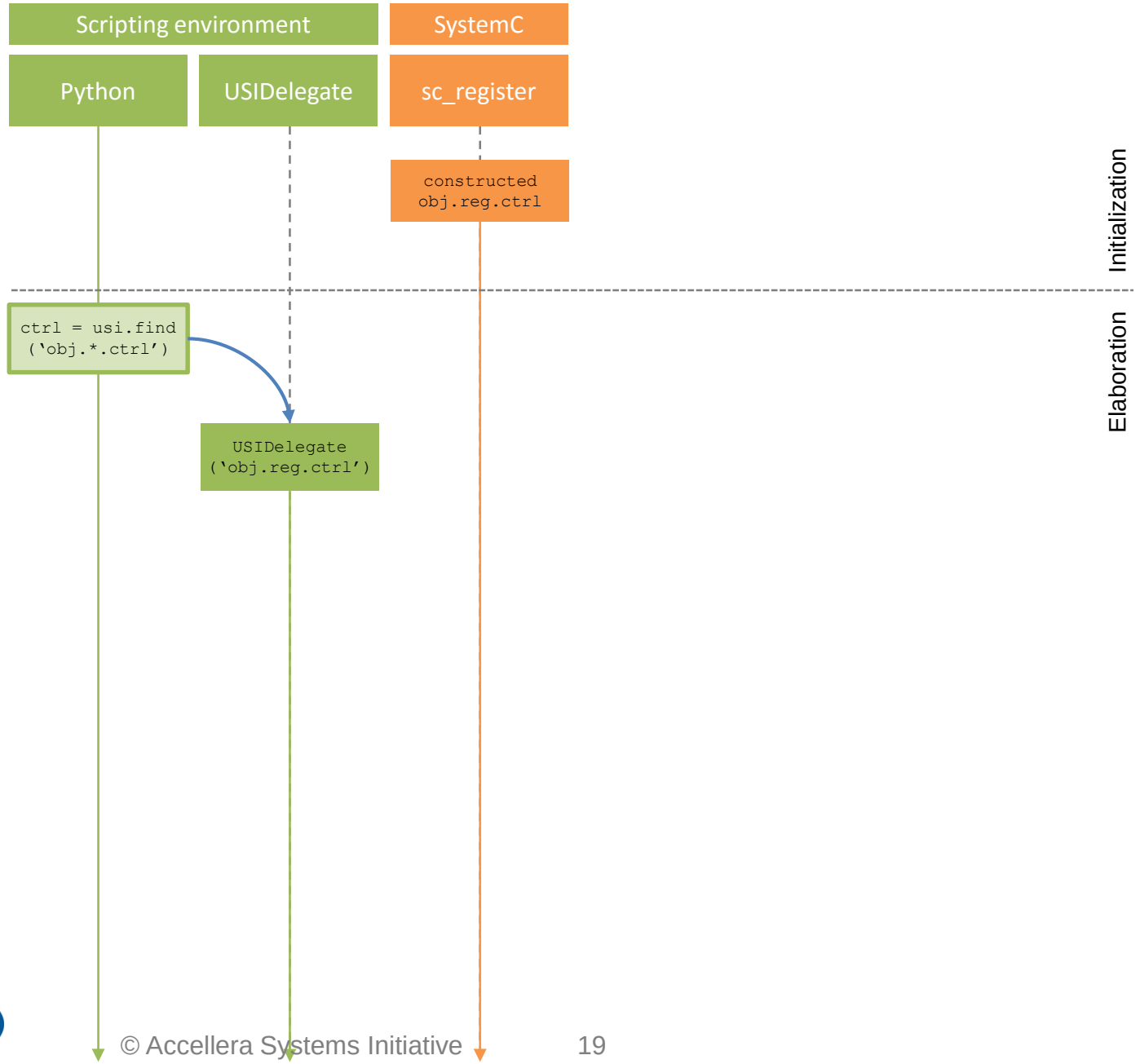
Scripting environment | SystemC

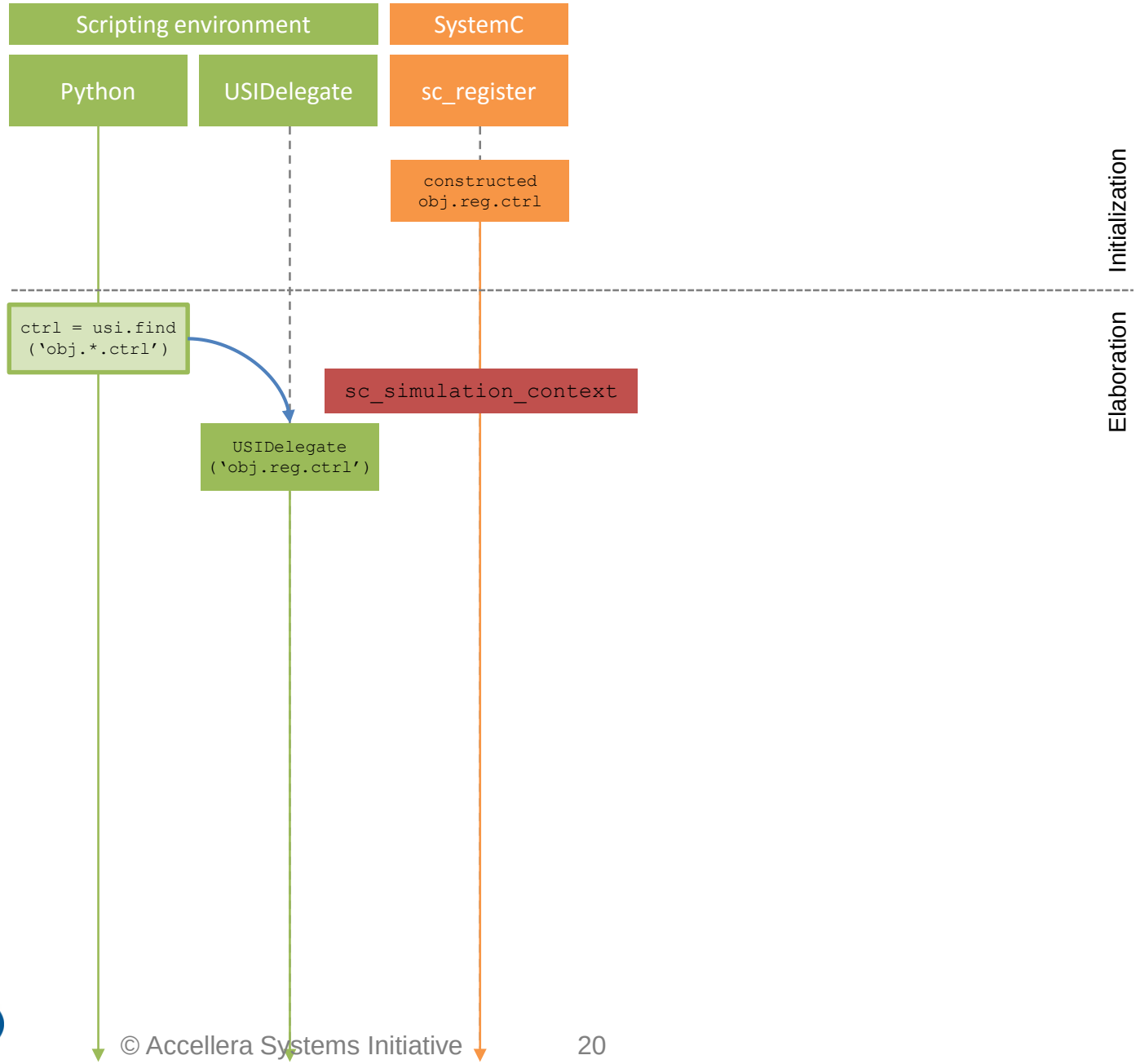

Scripting environment
SystemC
sc_register
constructed
obj.reg.ctrl

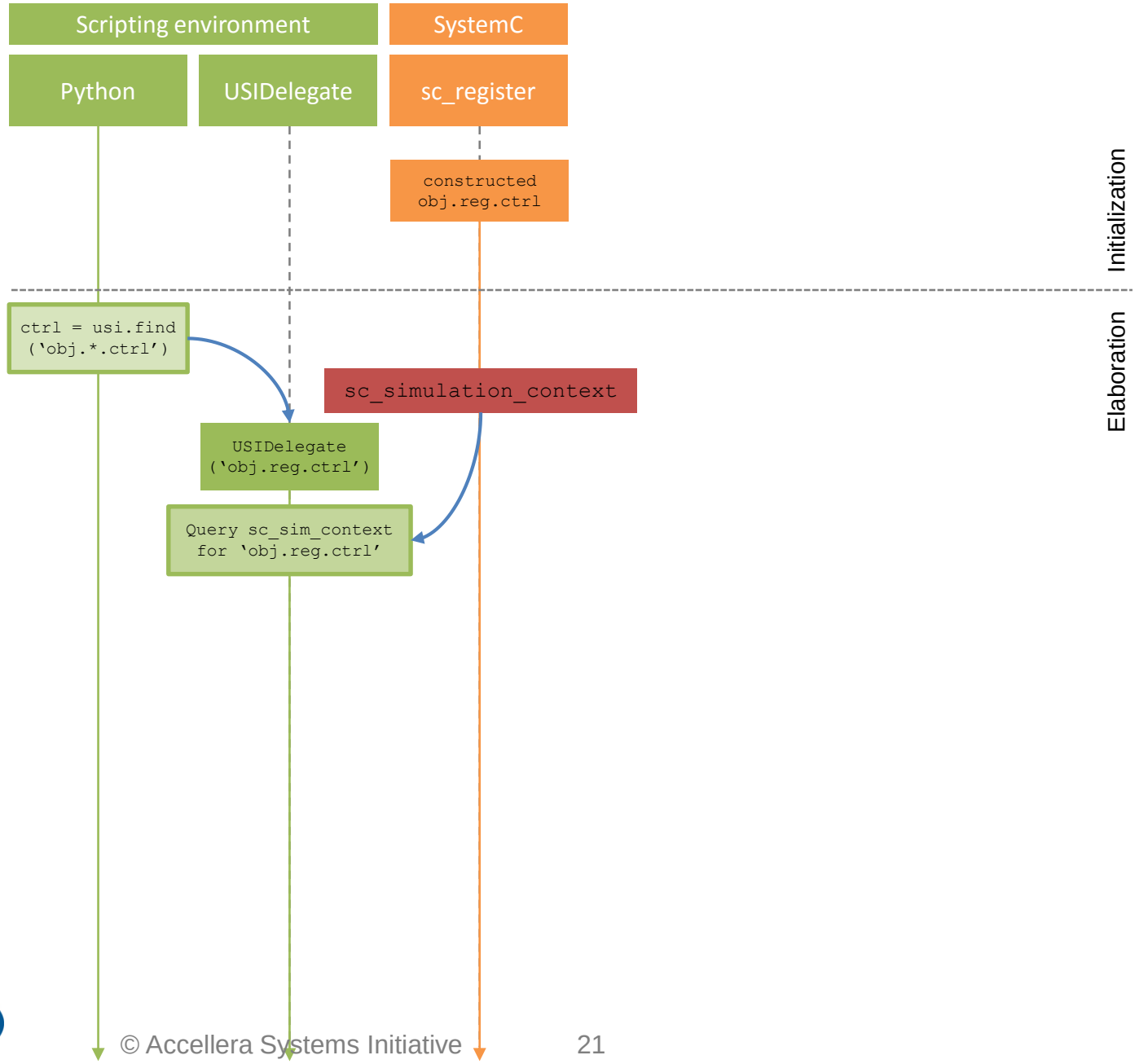


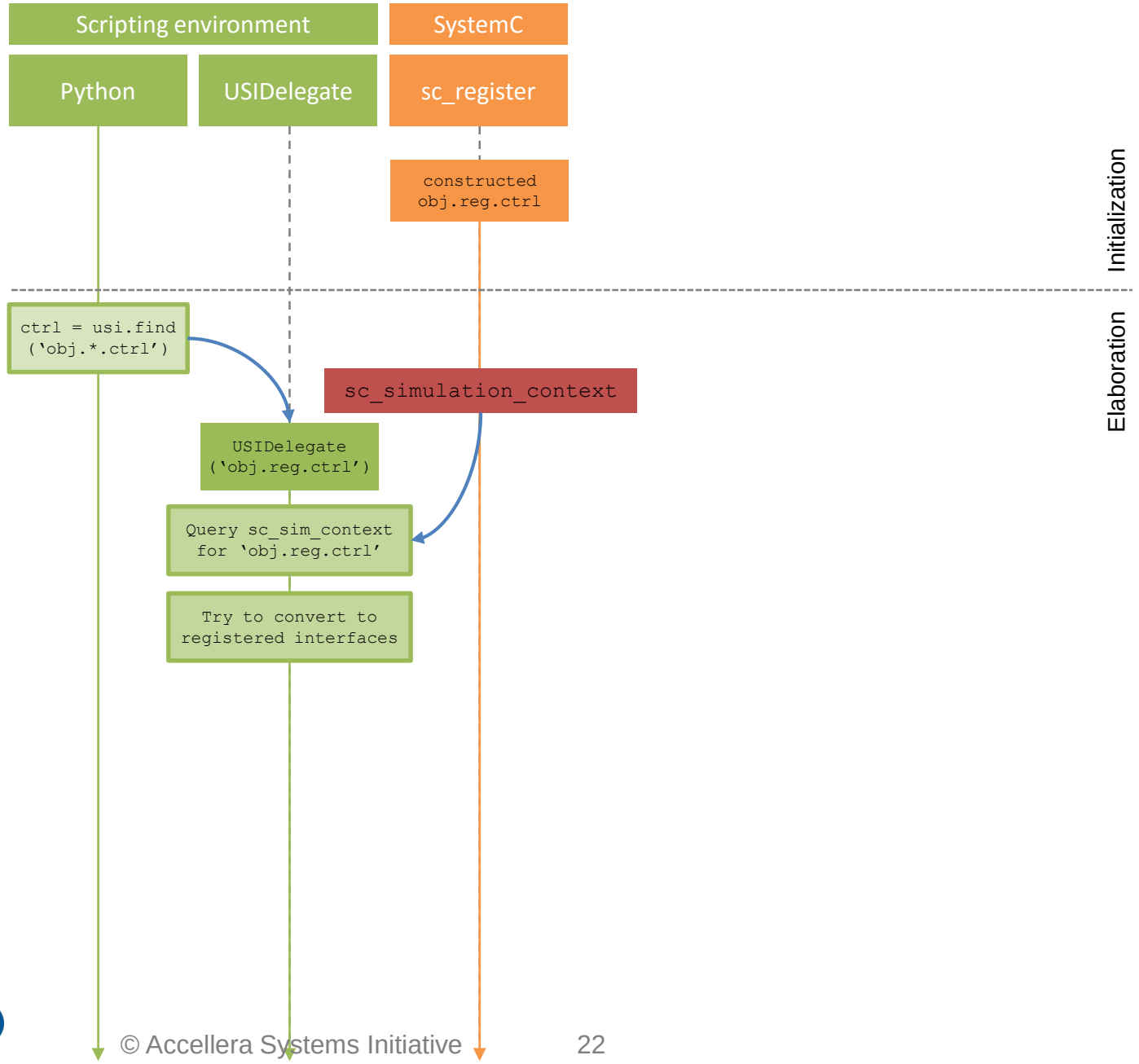

Scripting environment
SystemC
Python
sc_register
constructed
obj.reg.ctrl
Initialization
Elaboration

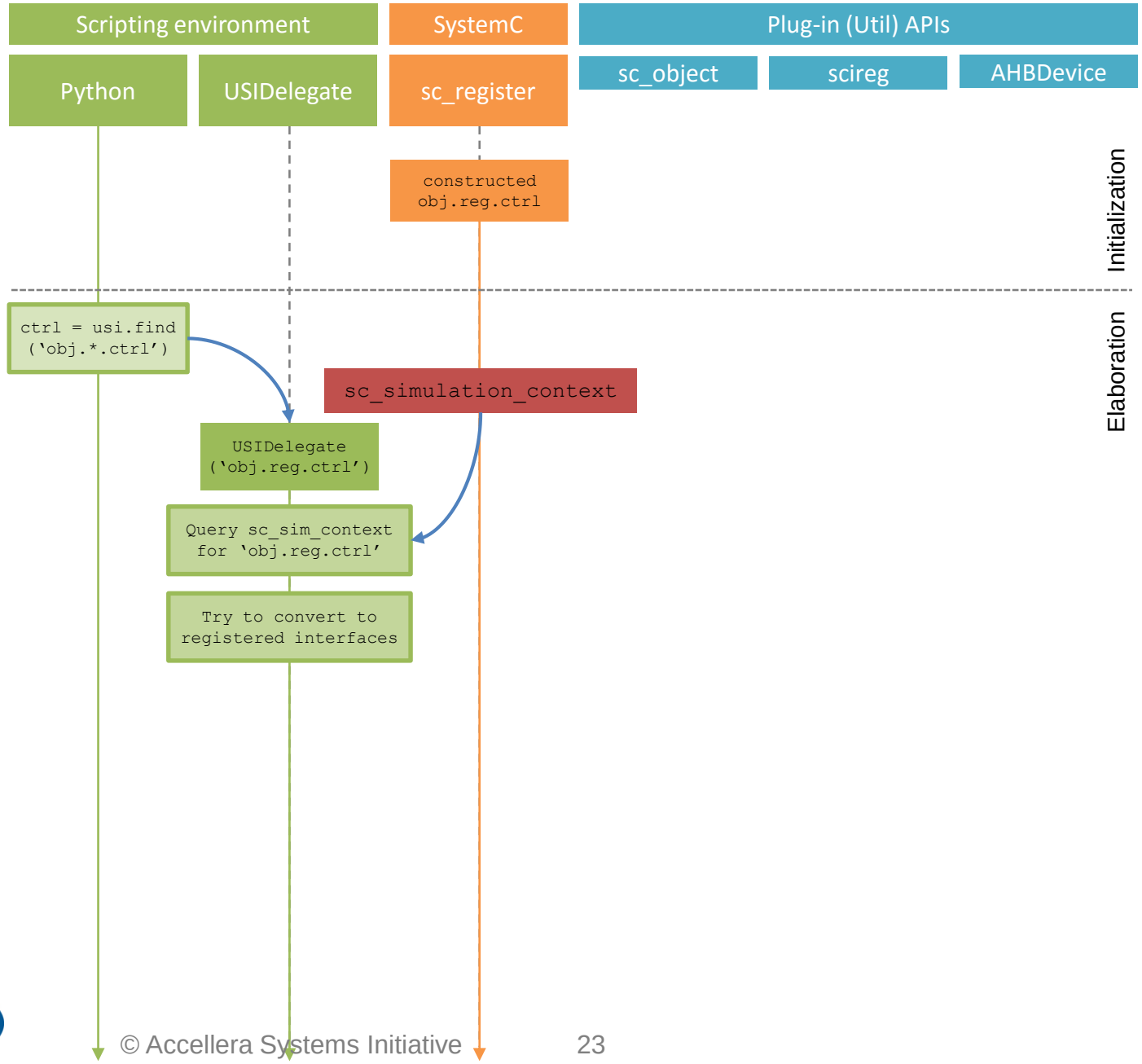

Scripting environment
SystemC
Python
sc_register
constructed
obj.reg.ctrl
Initialization
ctrl = usi.find
('obj.*.ctrl')
Elaboration

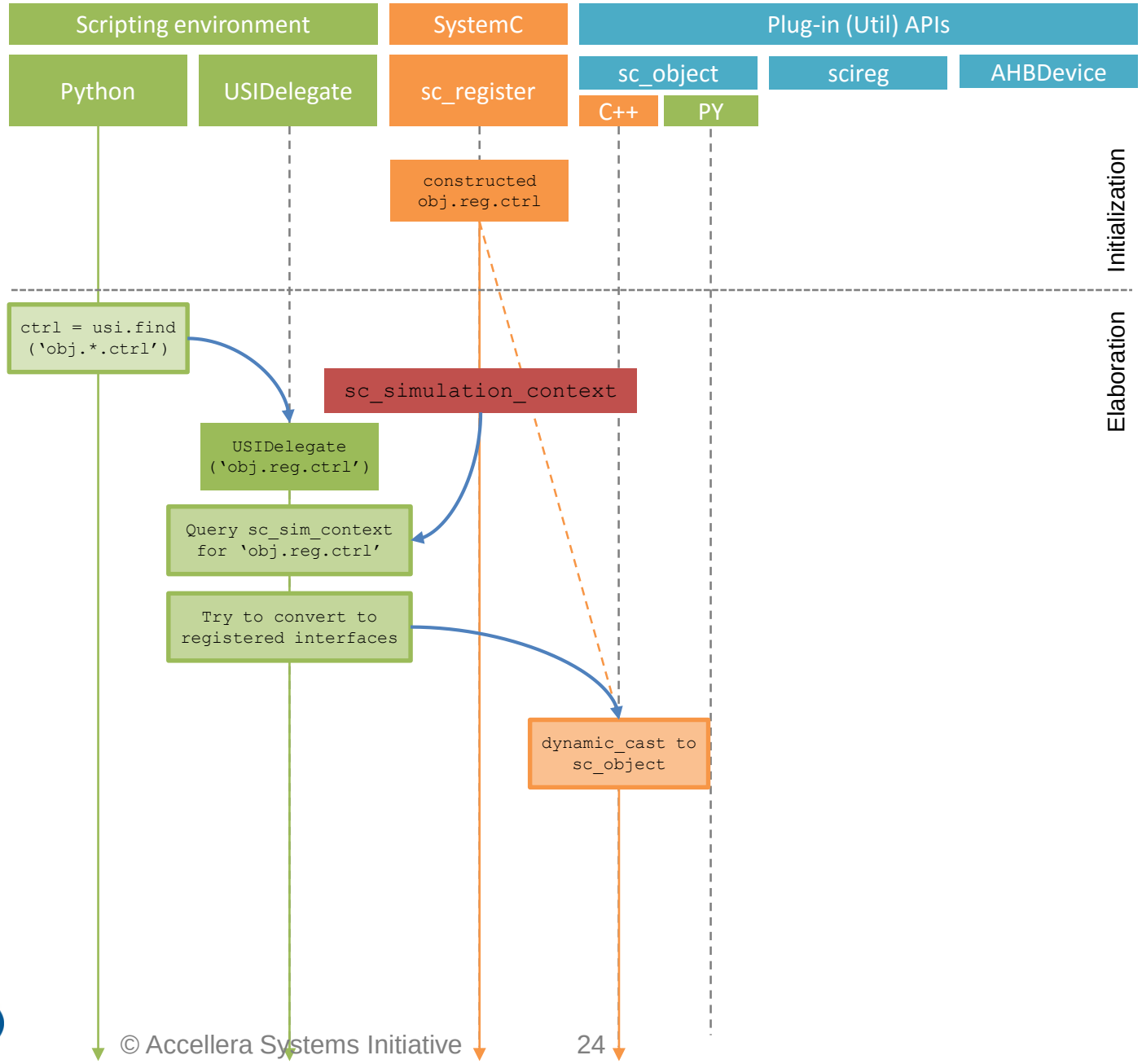

Scripting environment
SystemC
Python
USIDelegate
sc_register
constructed
obj.reg.ctrl
Initialization
Elaboration
ctrl = usi.find
('obj.*.ctrl')
USIDelegate
('obj.reg.ctrl')

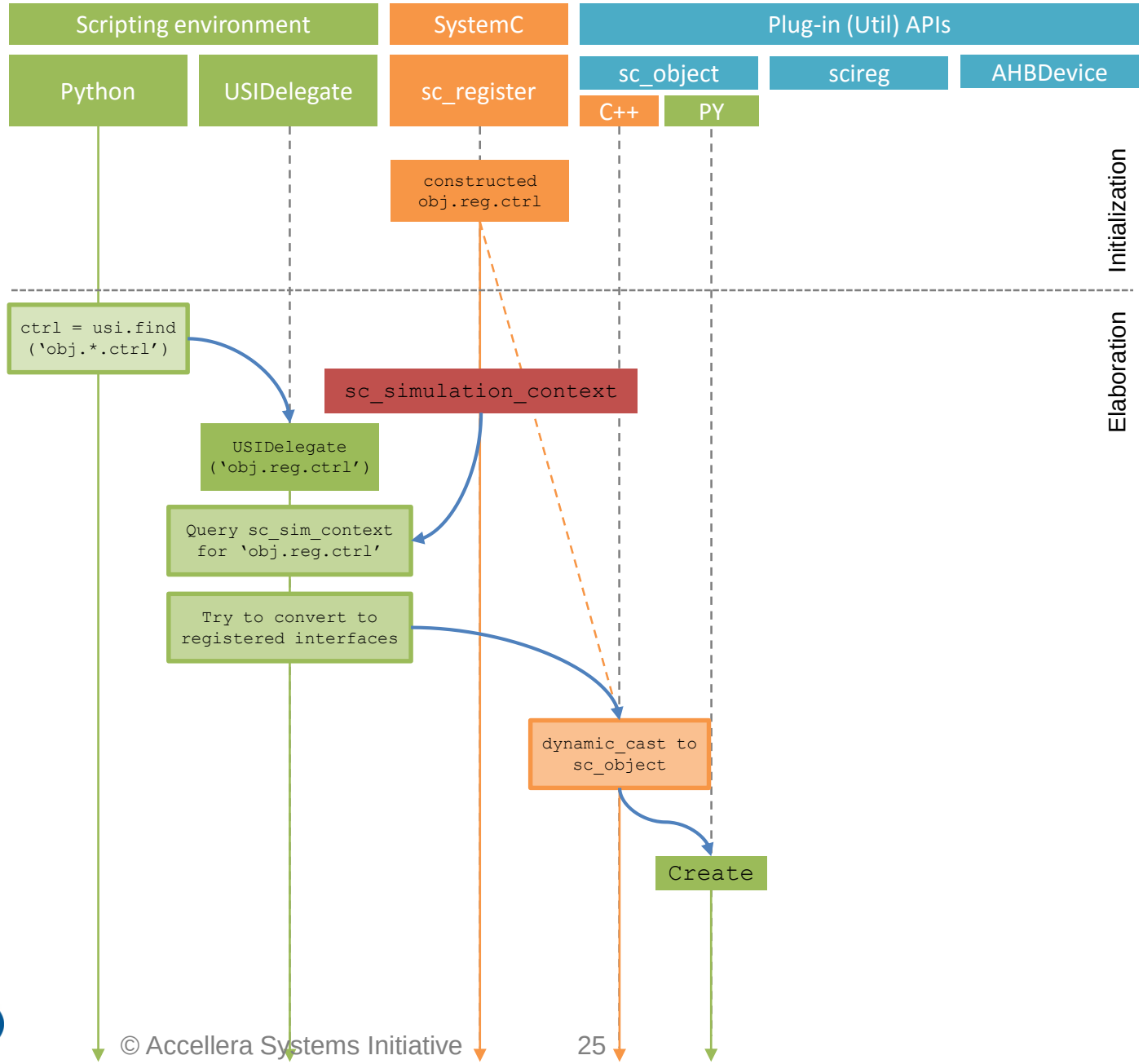

Scripting environment
SystemC
Python
USIDelegate
sc_register
constructed
obj.reg.ctrl
Initialization
Elaboration
ctrl = usi.find
('obj.*.ctrl')
sc_simulation_context
USIDelegate
('obj.reg.ctrl')

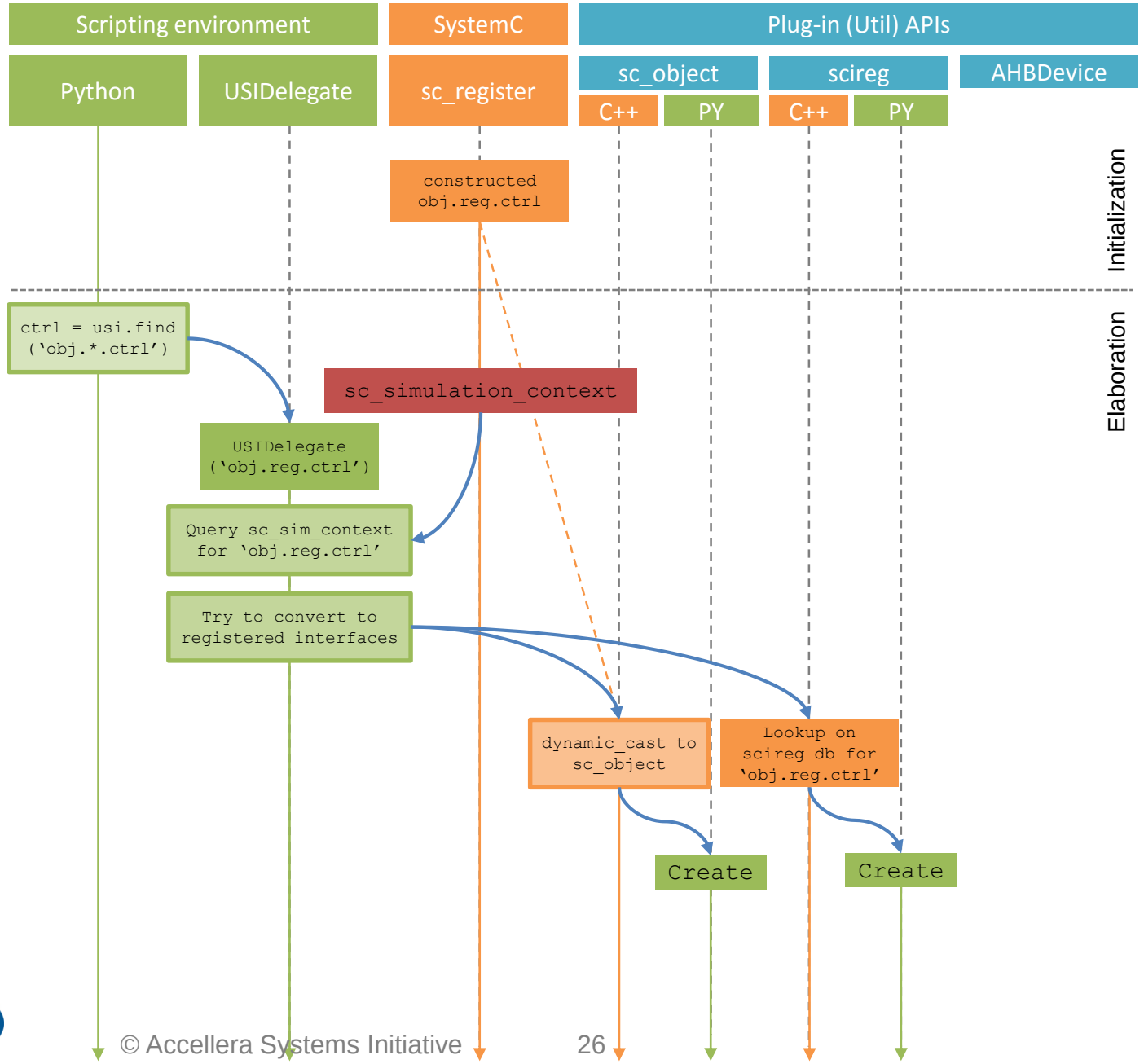

Scripting environment
SystemC
Python
USIDelegate
sc_register
constructed
obj.reg.ctrl
Initialization
Elaboration
ctrl = usi.find
('obj.*.ctrl')
sc_simulation_context
USIDelegate
('obj.reg.ctrl')
Query sc_sim_context
for 'obj.reg.ctrl'

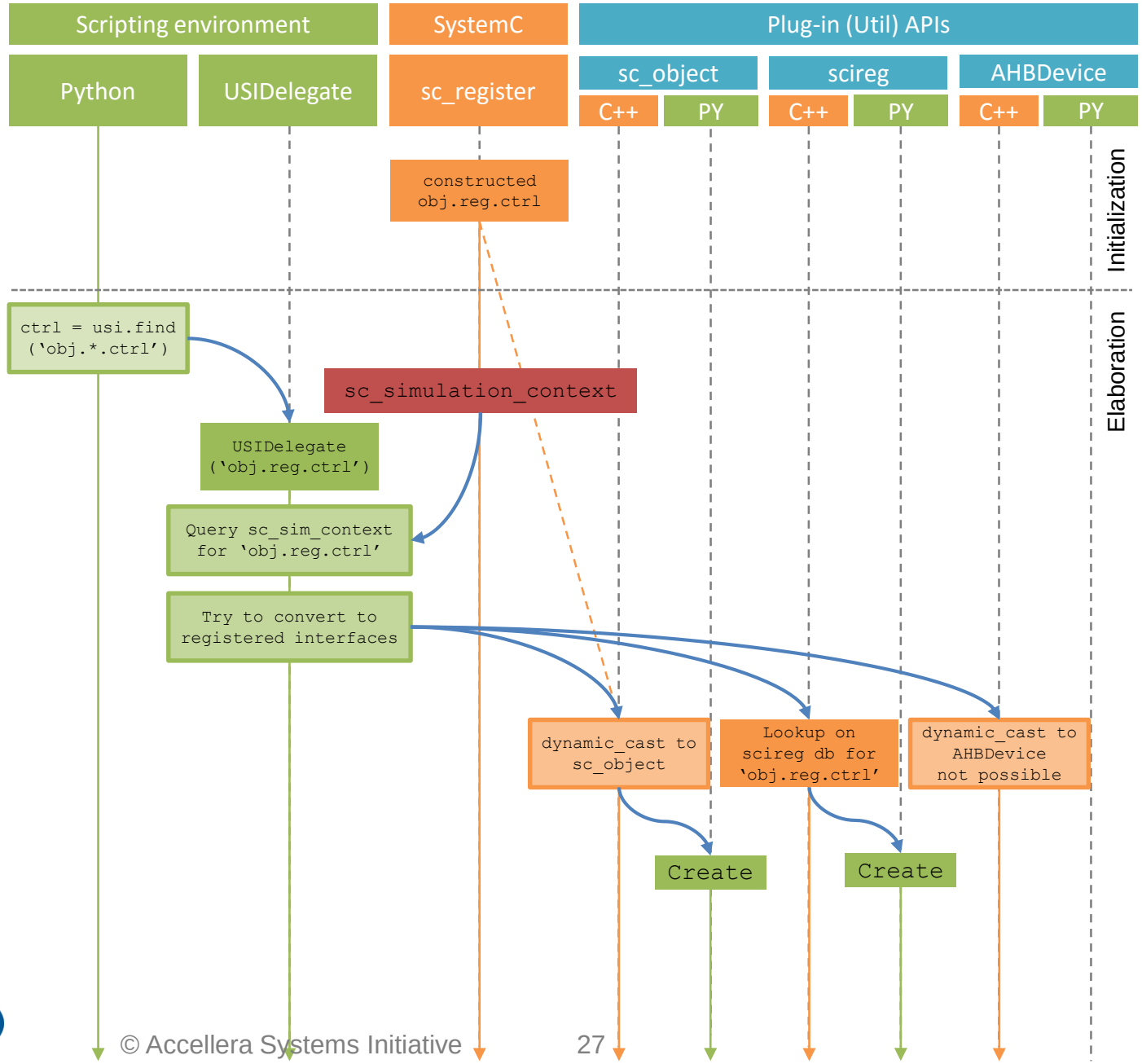

Scripting environment
SystemC
Python
USIDelegate
sc_register
constructed
obj.reg.ctrl
Initialization
Elaboration
ctrl = usi.find
('obj.*.ctrl')
sc_simulation_context
USIDelegate
('obj.reg.ctrl')
Query sc_sim_context
for 'obj.reg.ctrl'
Try to convert to
registered interfaces

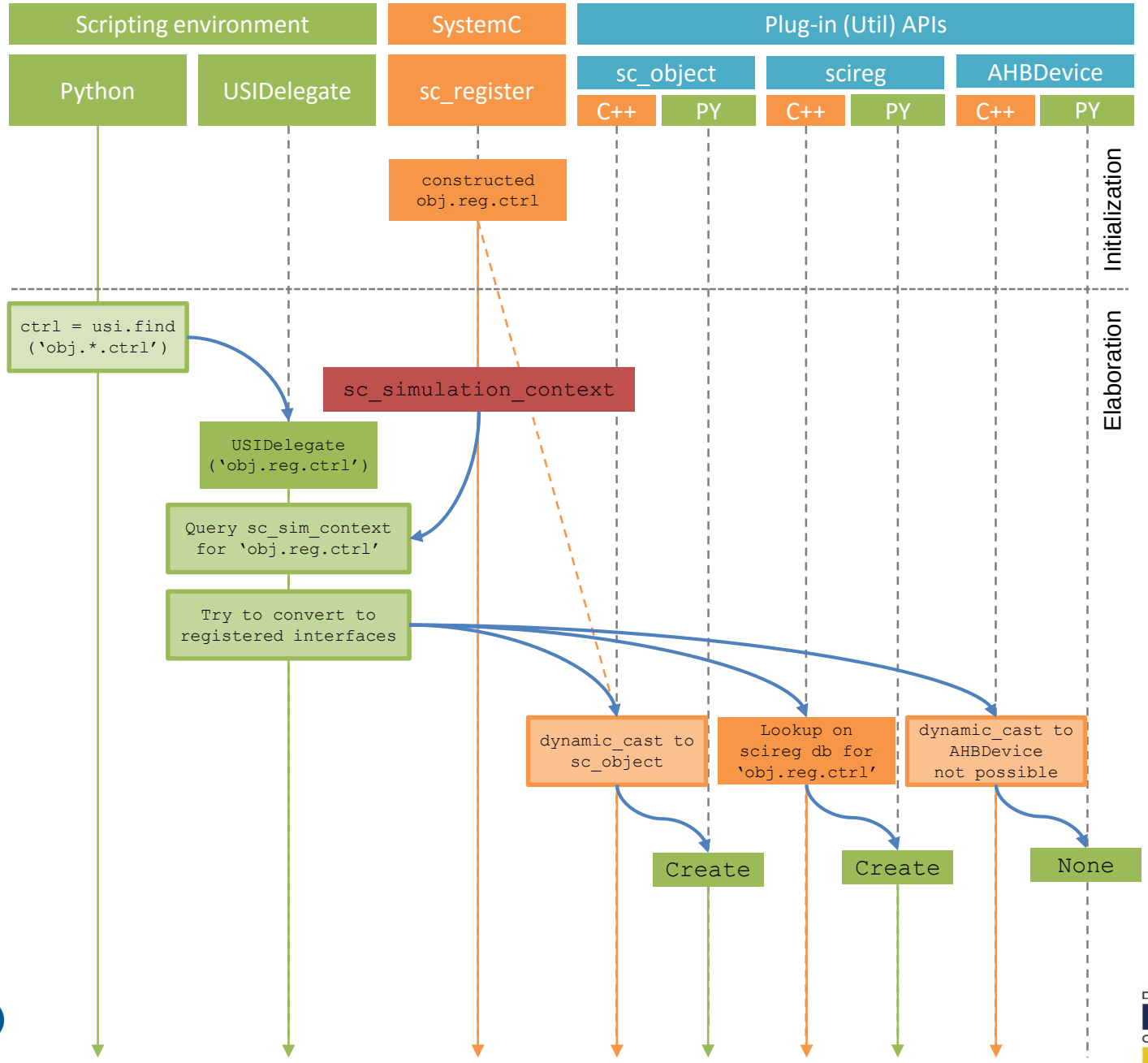

Scripting environment
SystemC
Plug-in (Util) APIs
Python
USIDelegate
sc_register
sc_object
scireg
AHBDevice
constructed
obj.reg.ctrl
Initialization
Elaboration
ctrl = usi.find
('obj.*.ctrl')
sc_simulation_context
USIDelegate
('obj.reg.ctrl')
Query sc_sim_context
for 'obj.reg.ctrl'
Try to convert to
registered interfaces

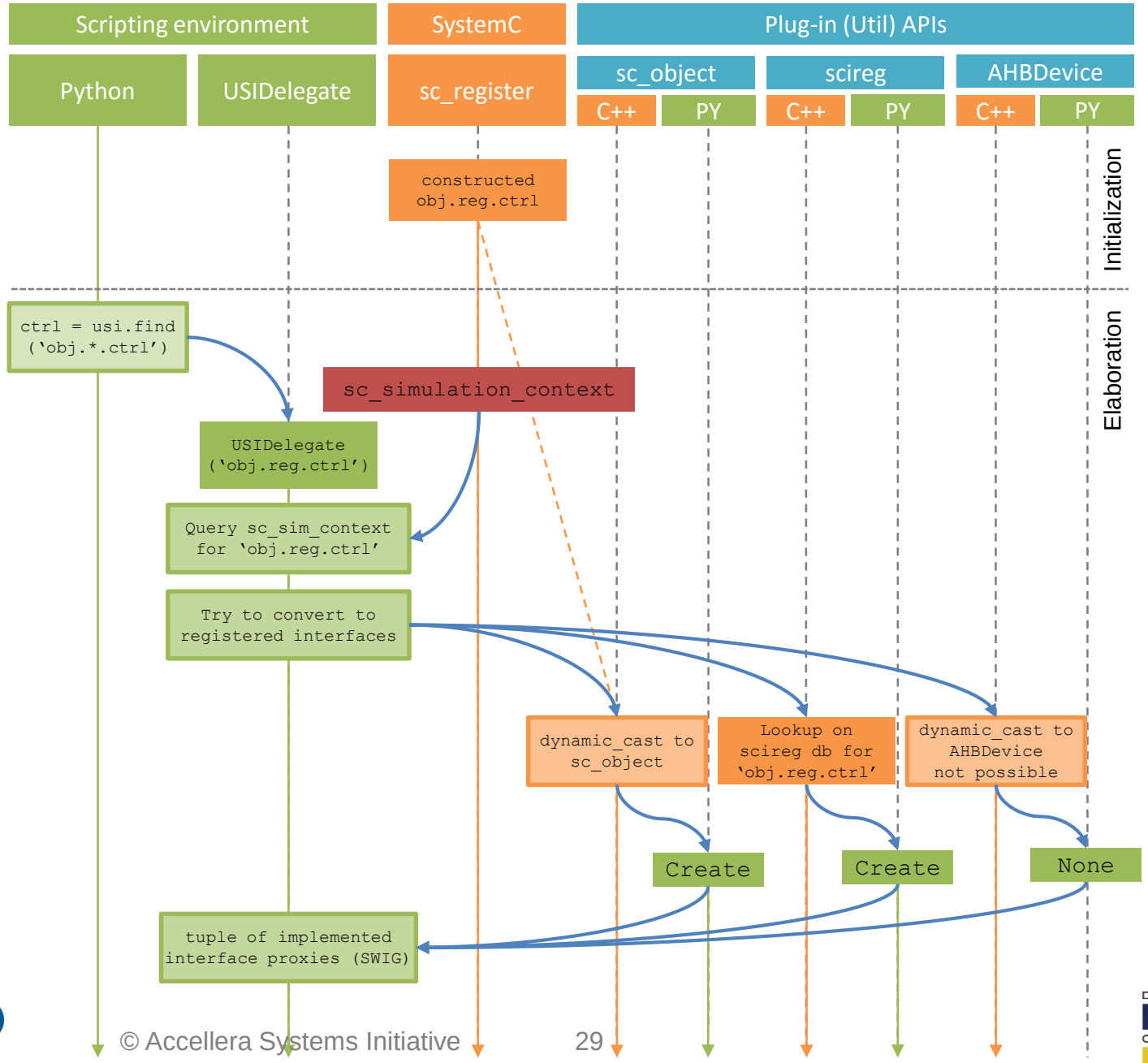

Scripting environment
SystemC
Plug-in (Util) APIs
Python
USIDelegate
sc_register
sc_object
scireg
AHBDevice
C++
PY
constructed
obj.reg.ctrl
Initialization
Elaboration
ctrl = usi.find
('obj.*.ctrl')
sc_simulation_context
USIDelegate
('obj.reg.ctrl')
Query sc_sim_context
for 'obj.reg.ctrl'
Try to convert to
registered interfaces
dynamic_cast to
sc_object


Scripting environment
SystemC
Plug-in (Util) APIs
Python
USIDelegate
sc_register
sc_object
scireg
AHBDevice
C++
PY
constructed
obj.reg.ctrl
Initialization
Elaboration
ctrl = usi.find
('obj.*.ctrl')
sc_simulation_context
USIDelegate
('obj.reg.ctrl')
Query sc_sim_context
for 'obj.reg.ctrl'
Try to convert to
registered interfaces
dynamic_cast to
sc_object
Create



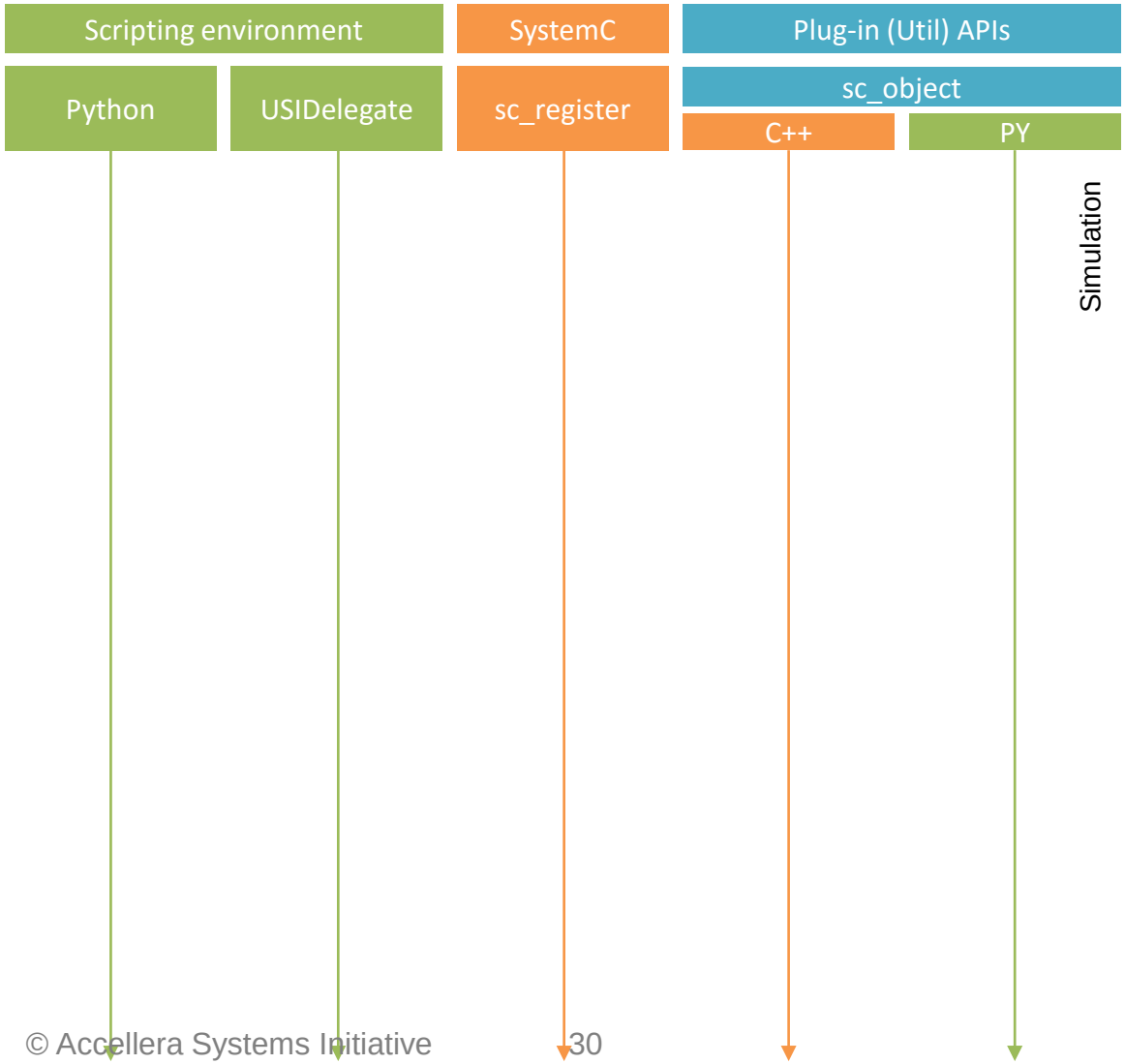






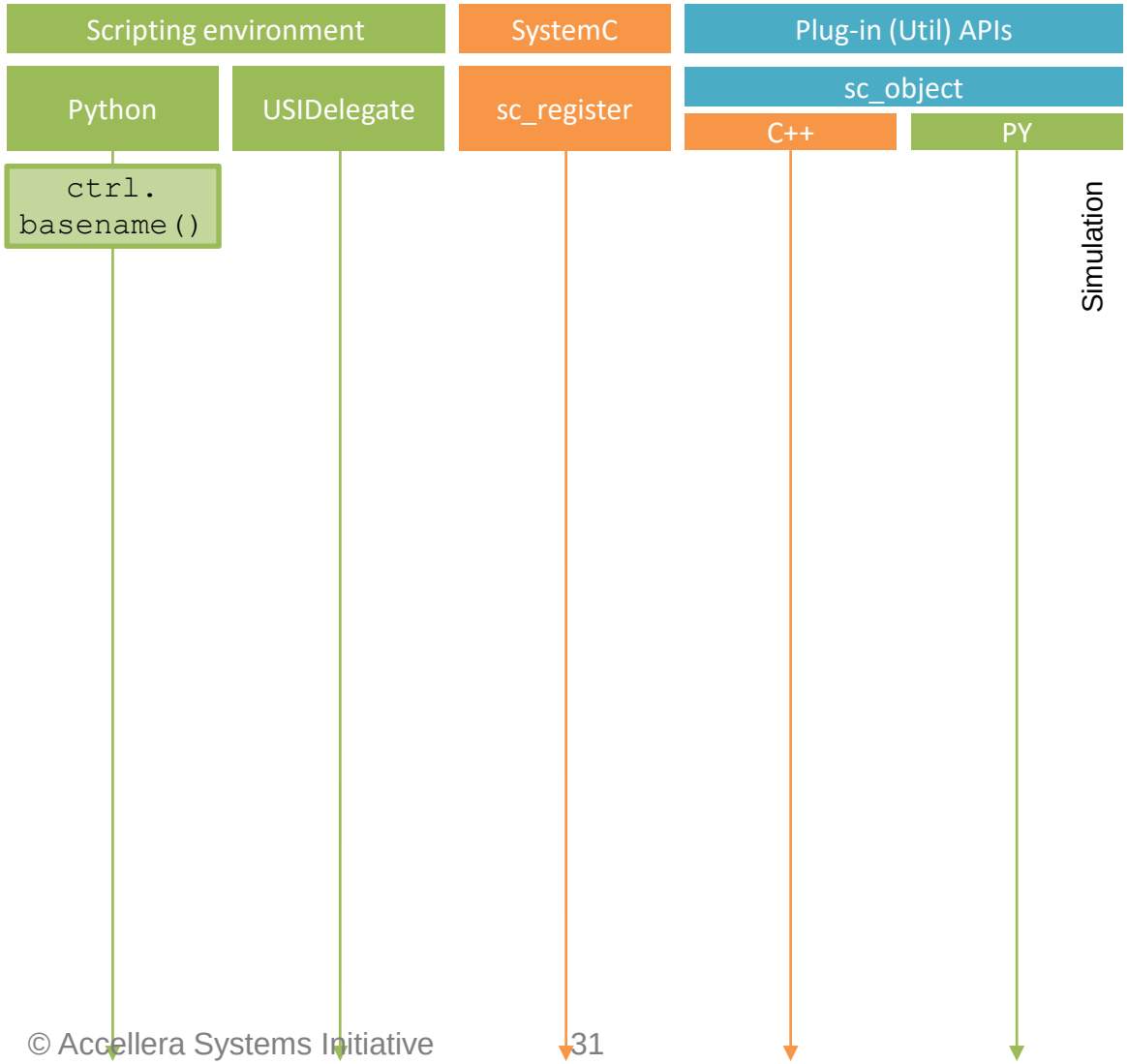
Plugin function call

Or how to access simulation information via interface delegation



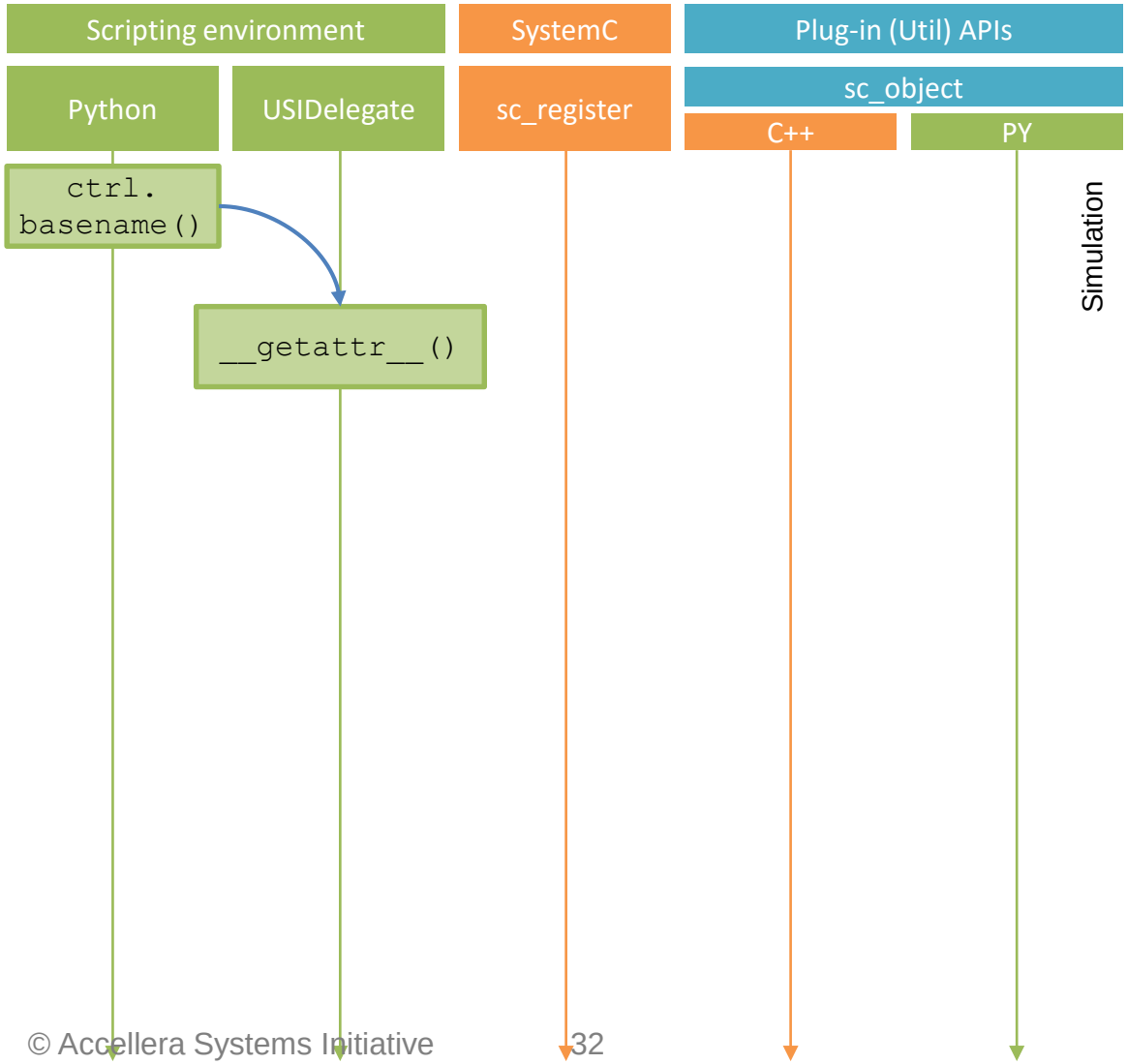
Plugin function call

Or how to access simulation information via interface delegation



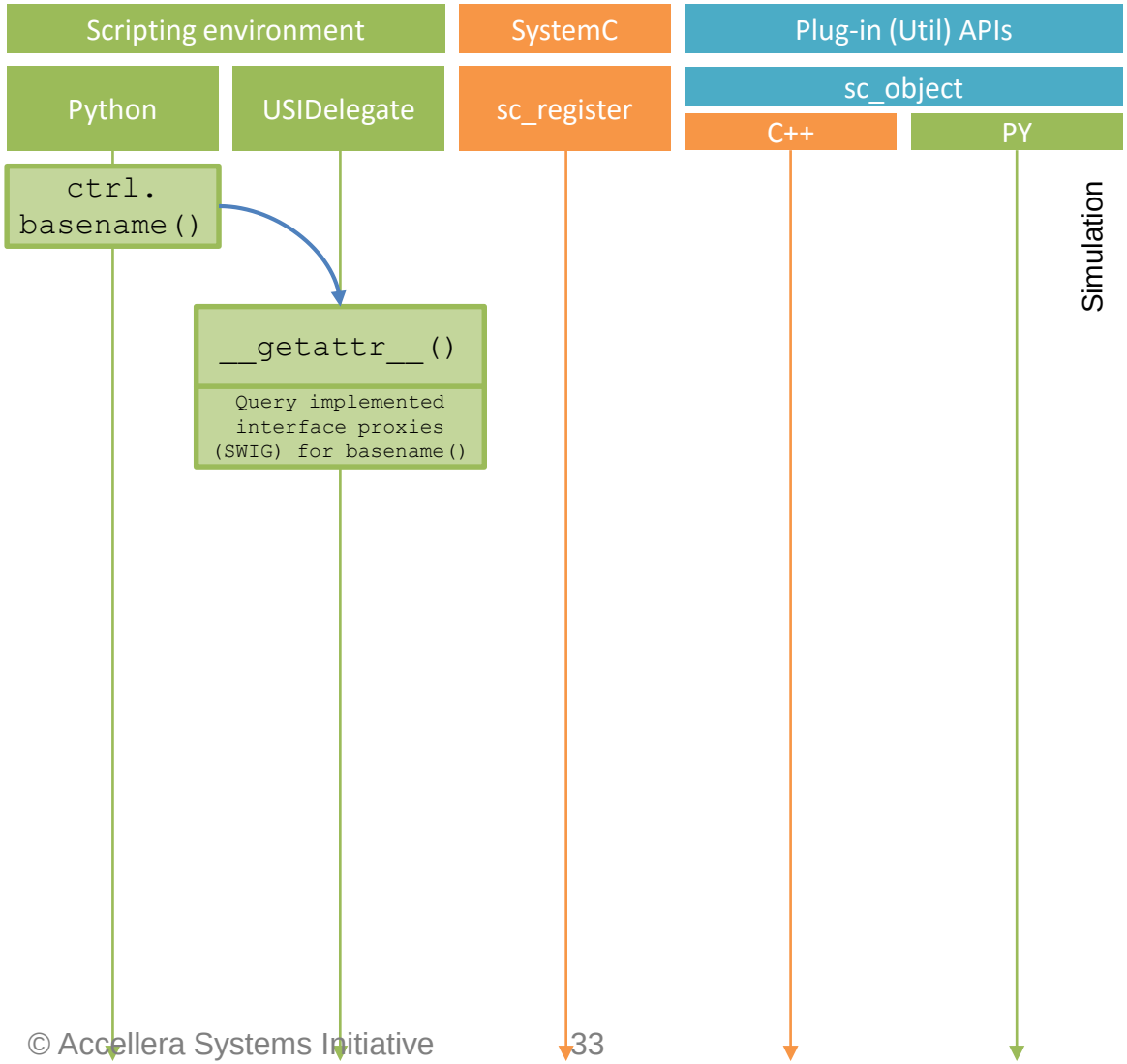
Plugin function call

Or how to access simulation information via interface delegation



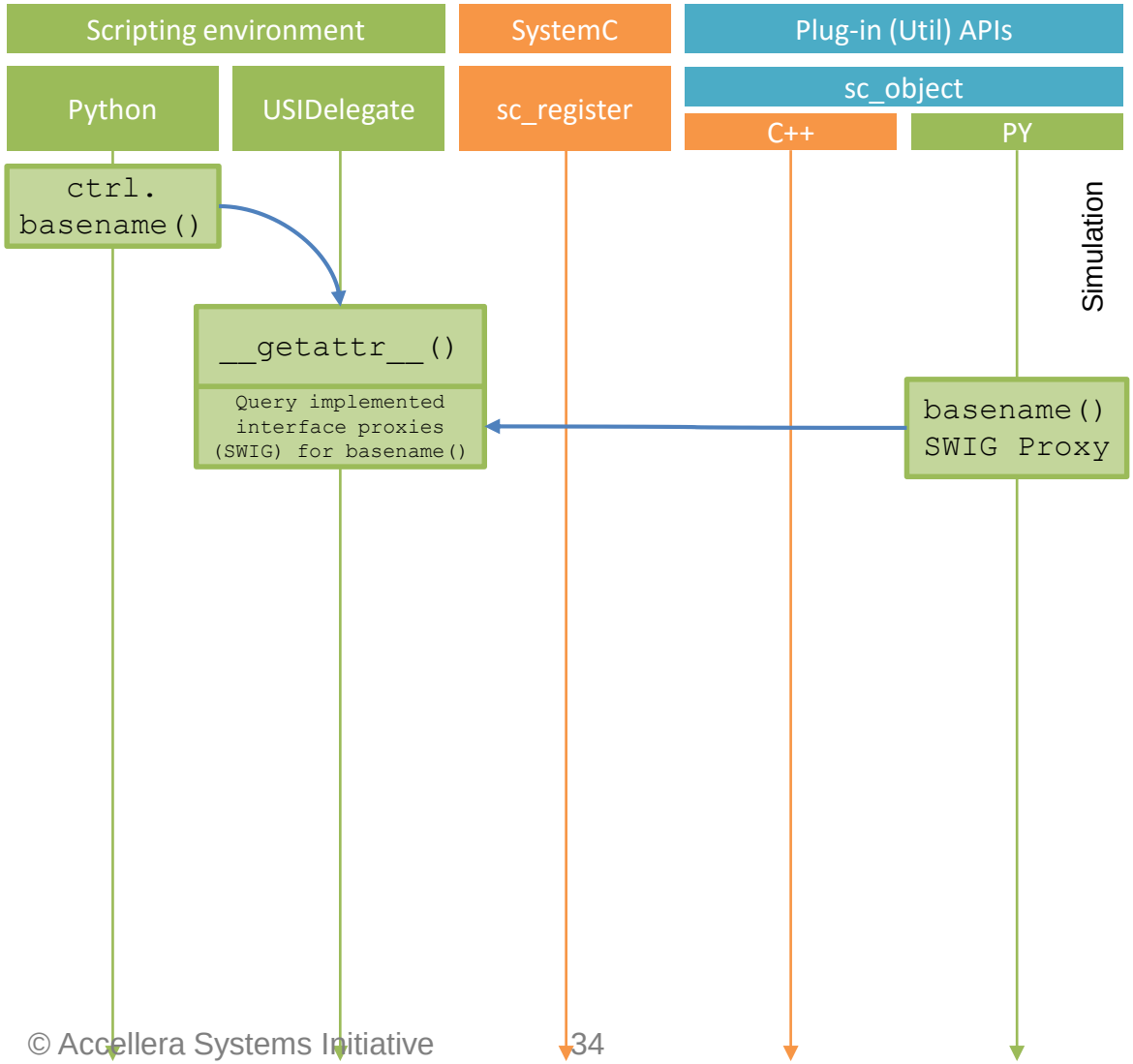
Plugin function call

Or how to access simulation information via interface delegation



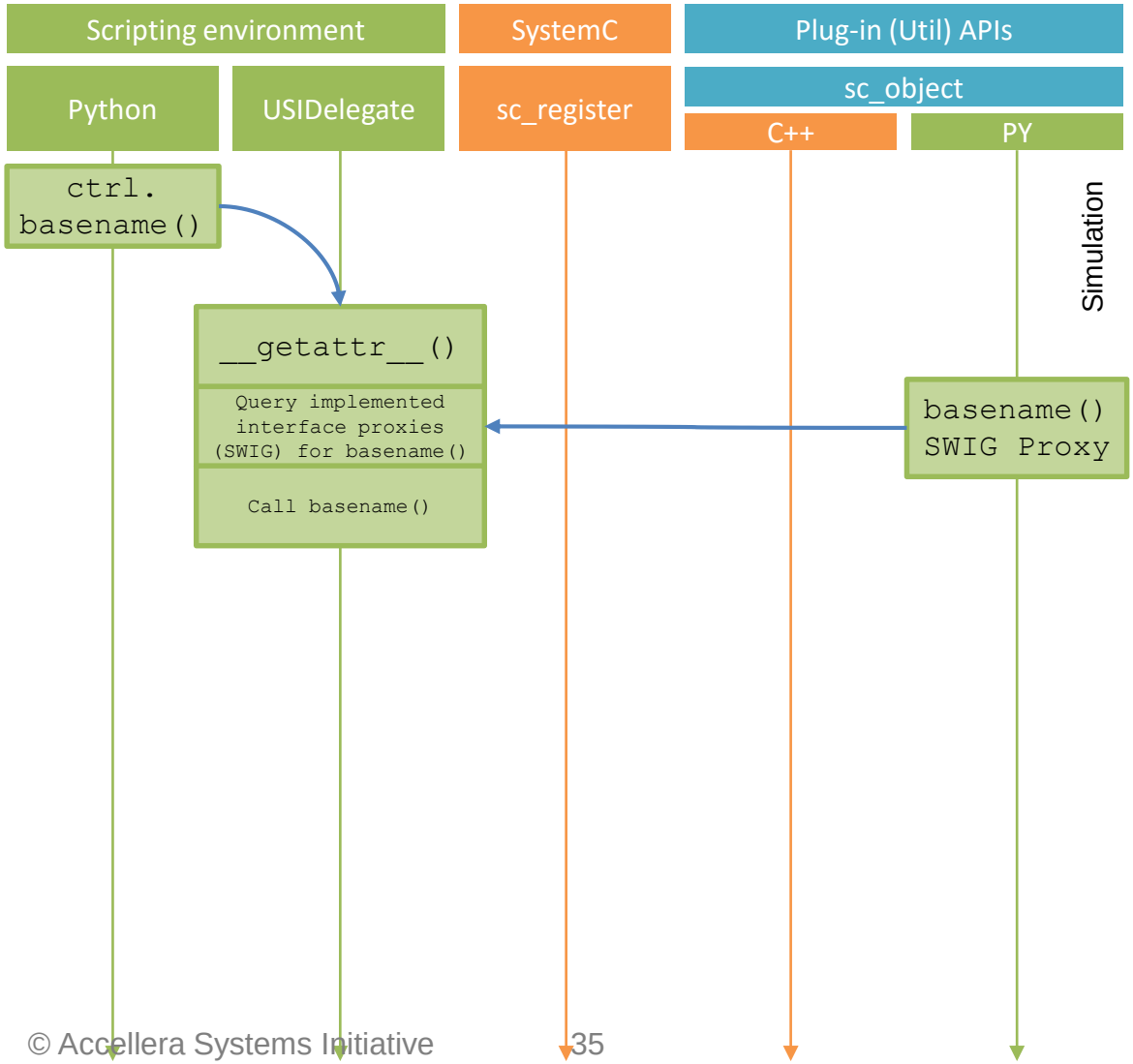
Plugin function call

Or how to access simulation information via interface delegation



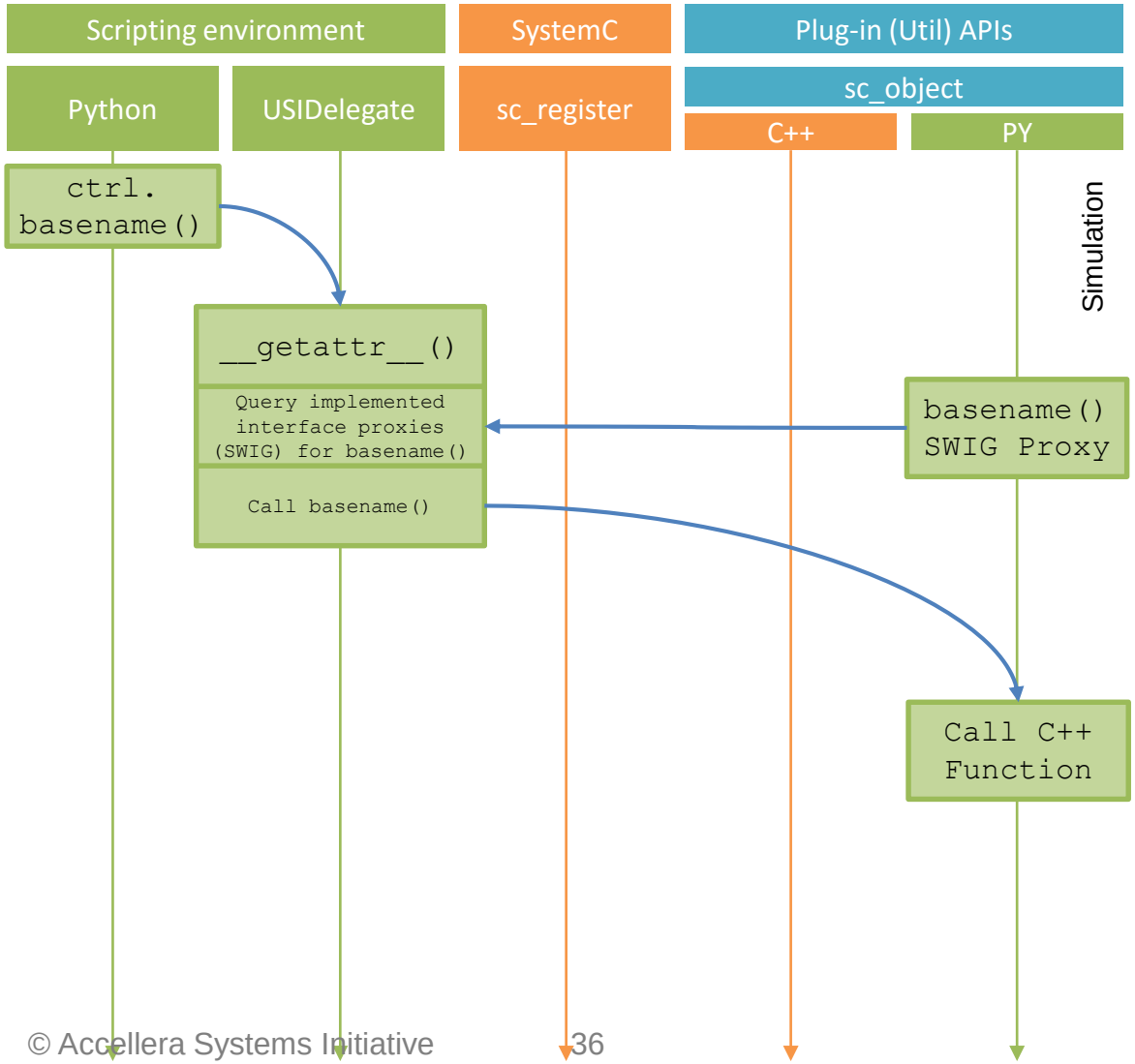
Plugin function call

Or how to access simulation information via interface delegation



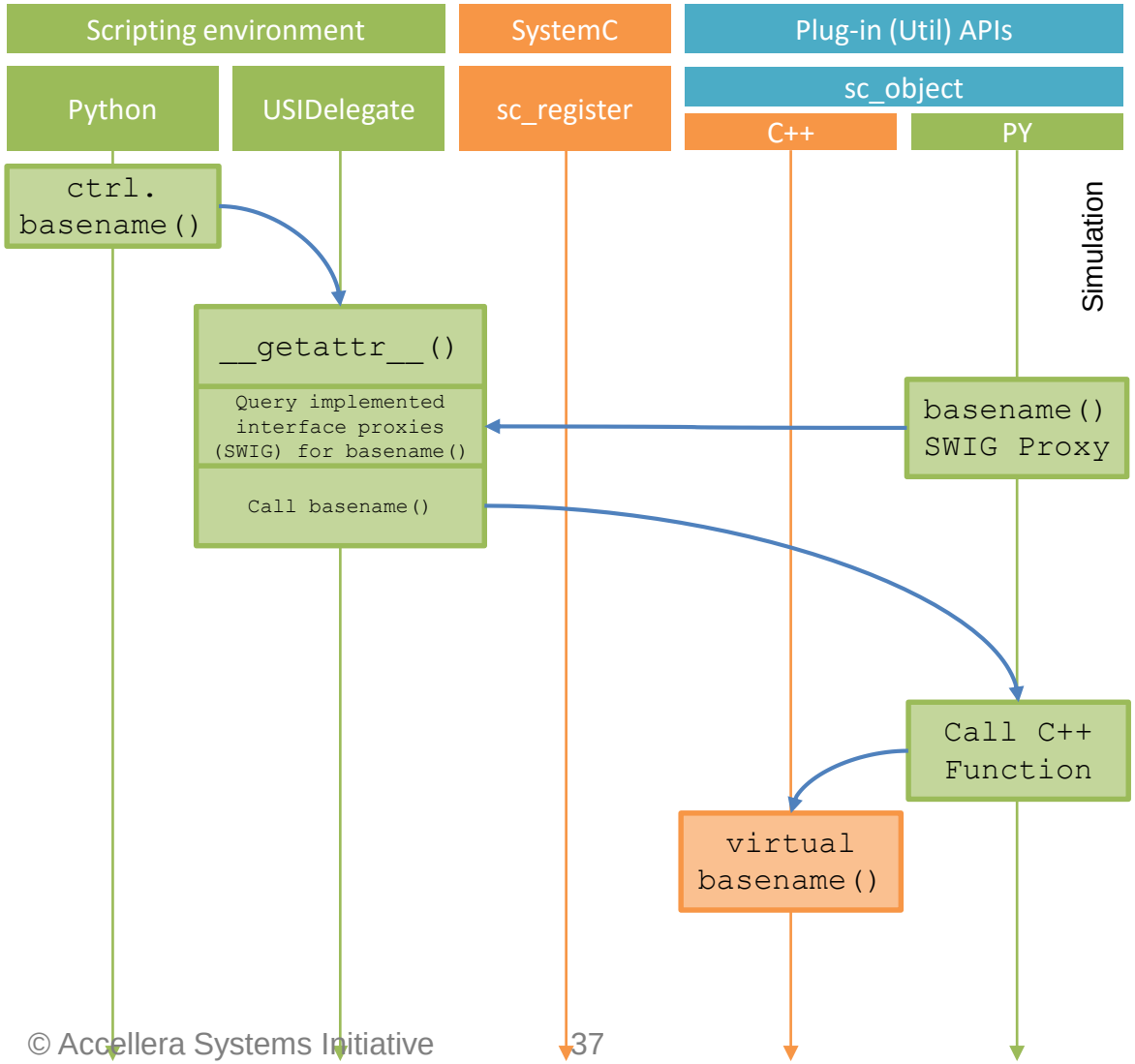
Plugin function call

Or how to access simulation information via interface delegation



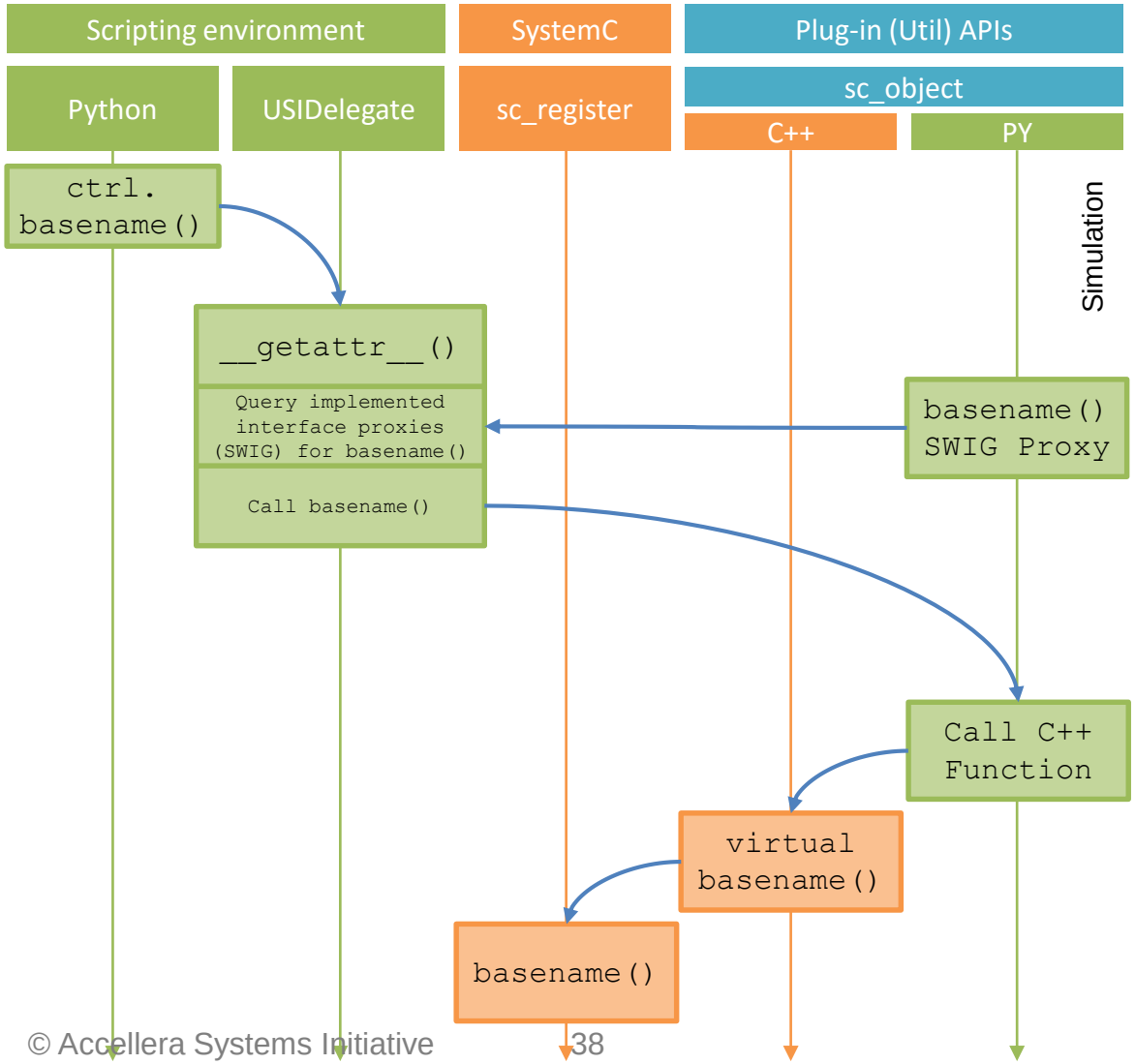
Plugin function call

Or how to access simulation information via interface delegation



Plugin function call

Or how to access simulation information via interface delegation



Simulator agnostic

- ✓ Accellera SystemC
- ✓ Mentor Graphics Questasim
  - No sc_main
  - Introduction of a help macro
  - USI_MODEL_EXPORT
  - GCC 32 bit
  - Debugging problems
  - Potential for deeper integration

Language agnostic

- ✓ TCL
- ✓ Ruby
- ✓ Python
- ○ Lua

API agnostic

- ✓ sc_object, tlm_socket
- ✓ GreenControl gs_param
- ✓ Cadence scireg
- ✓ AHB/APBDevice

Usability vs. Speed

- Easier to write
- Faster prototyping
- Domain Specific
  - Statistics
  - Text processing

- Slower (Python: 10 to 100 times)
- Fast extensions available (e.g. Python pandas)
- Ideal for pre-/post processing and glue

SoCRocket is available online: https://socrocket.github.io/

For more information please contact us!

Questions