

# OOA of MASL

[xtuml.org](http://xtuml.org)

[cortland.starrett@onefact.net](mailto:cortland.starrett@onefact.net)



OOA of MASL

# outline

- goals
- strategy
- OOA of MASL
- MASL loader
- next steps

# goals

- capture MASL as a model
- parse and load correct MASL
- anticipate cloud architecture



# strategy

- port MASL model compiler java model
  - leverage working starting point
  - low-risk with large number of artifacts
  - script conversion of java to MASL
  - import into BridgePoint
- adjust ported model
- refine with action language

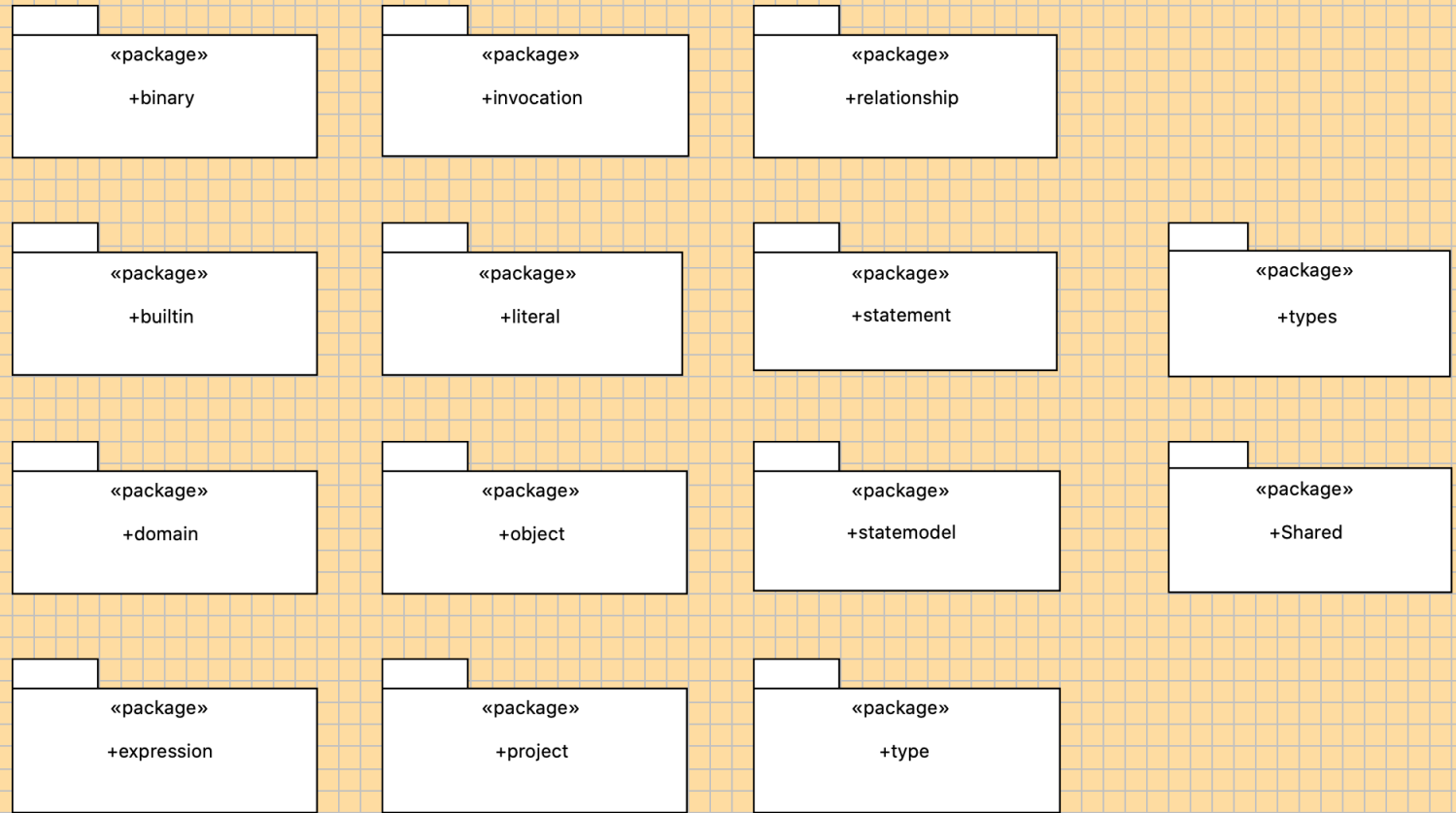


# OOA of MASL

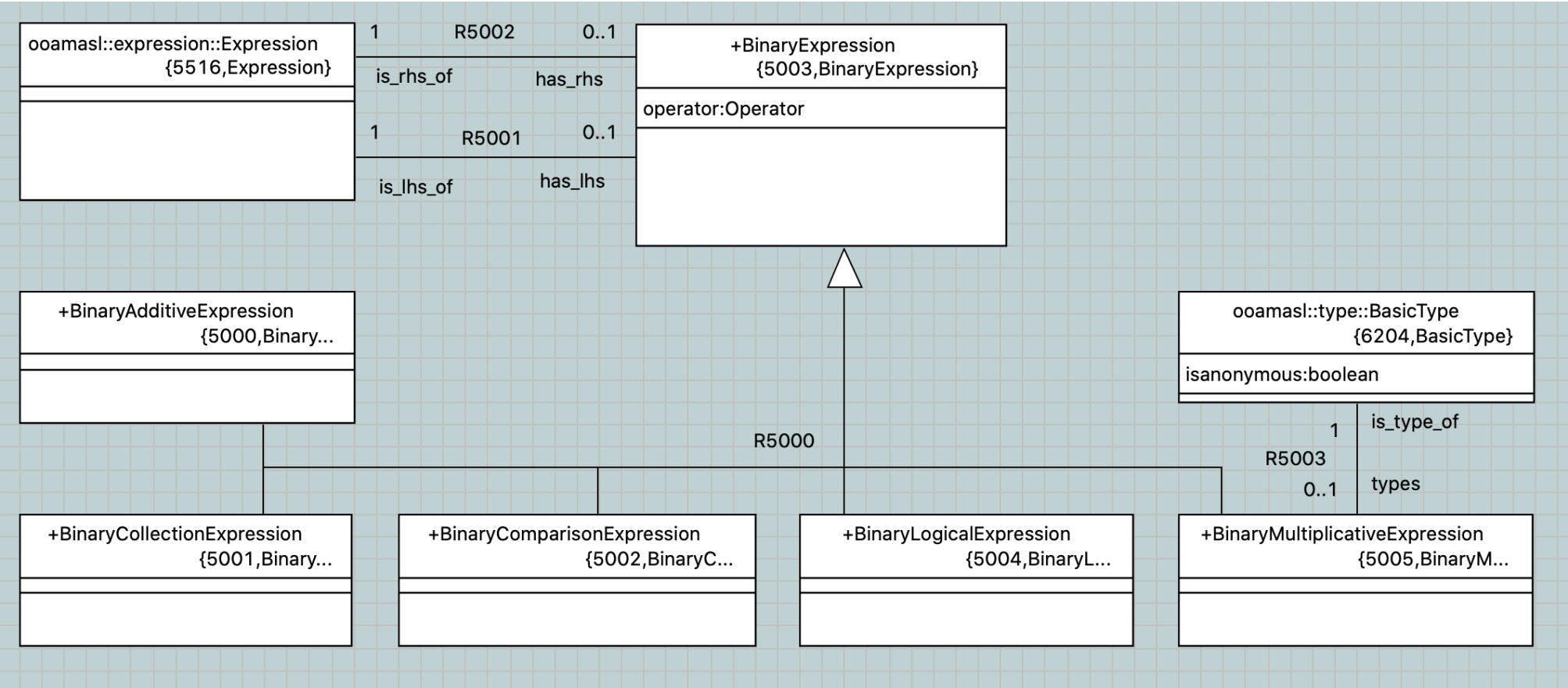
- ooamasl subsystem packages
- package references and *naked* OOA
- statistics
- deployment component (ME and canvas)
- tour of each subsystem

# OOA of MASL: ooamasl packages

- ▼ ooamasl [mc 11745\_loadmasl3]
  - ▶ binary
  - ▶ builtin
  - ▶ domain
  - ▶ expression
  - ▶ invocation
  - ▶ literal
  - ▶ object
  - ▶ project
  - ▶ relationship
  - ▶ Shared
  - ▶ statement
  - ▶ statemodel
  - ▶ type
  - ▶ types
- ▶ runtime [ciera master]
- ▶ > stratus [mc 11745\_loadmasl3]

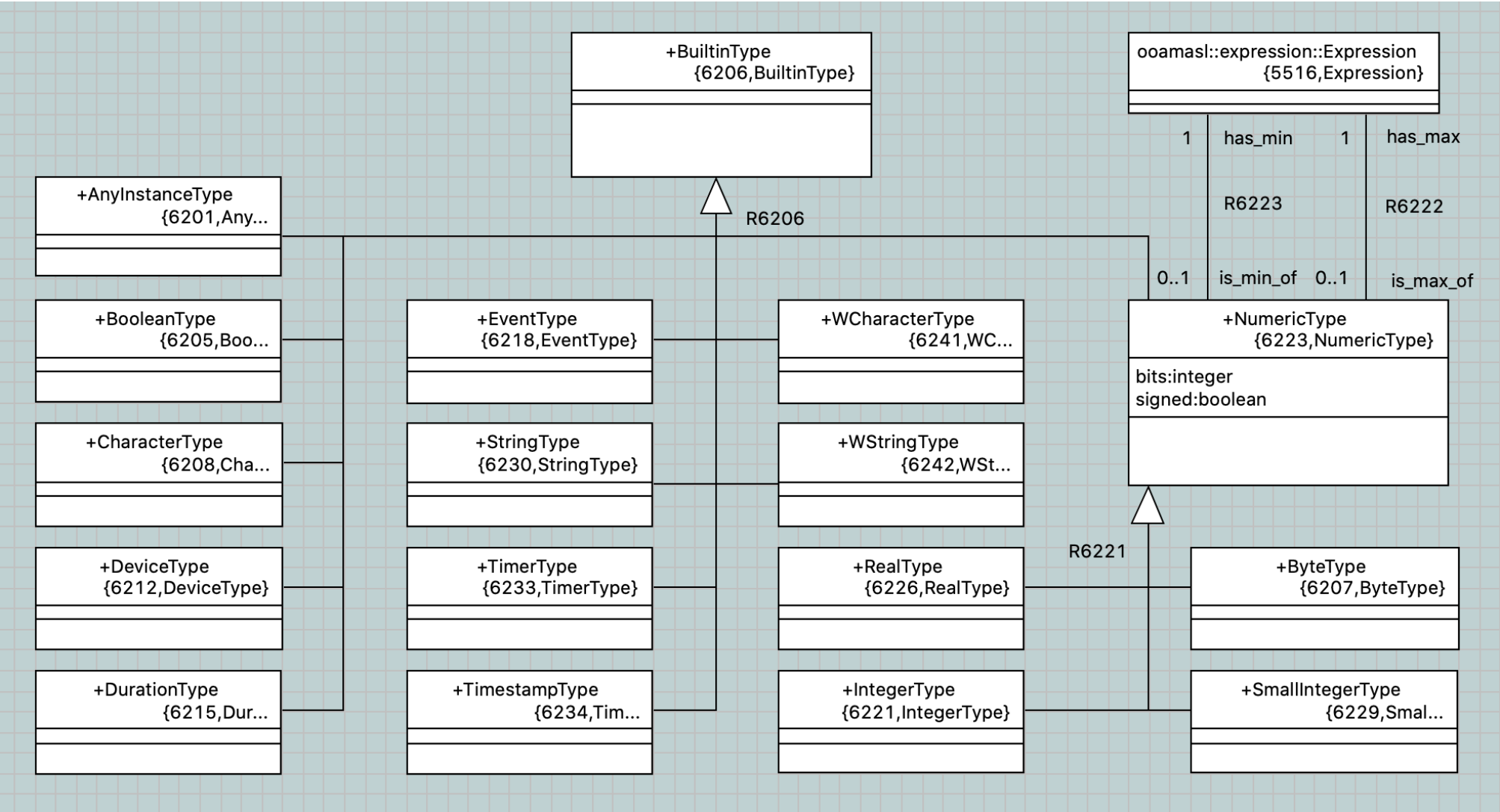


# OOA of MASL: binary

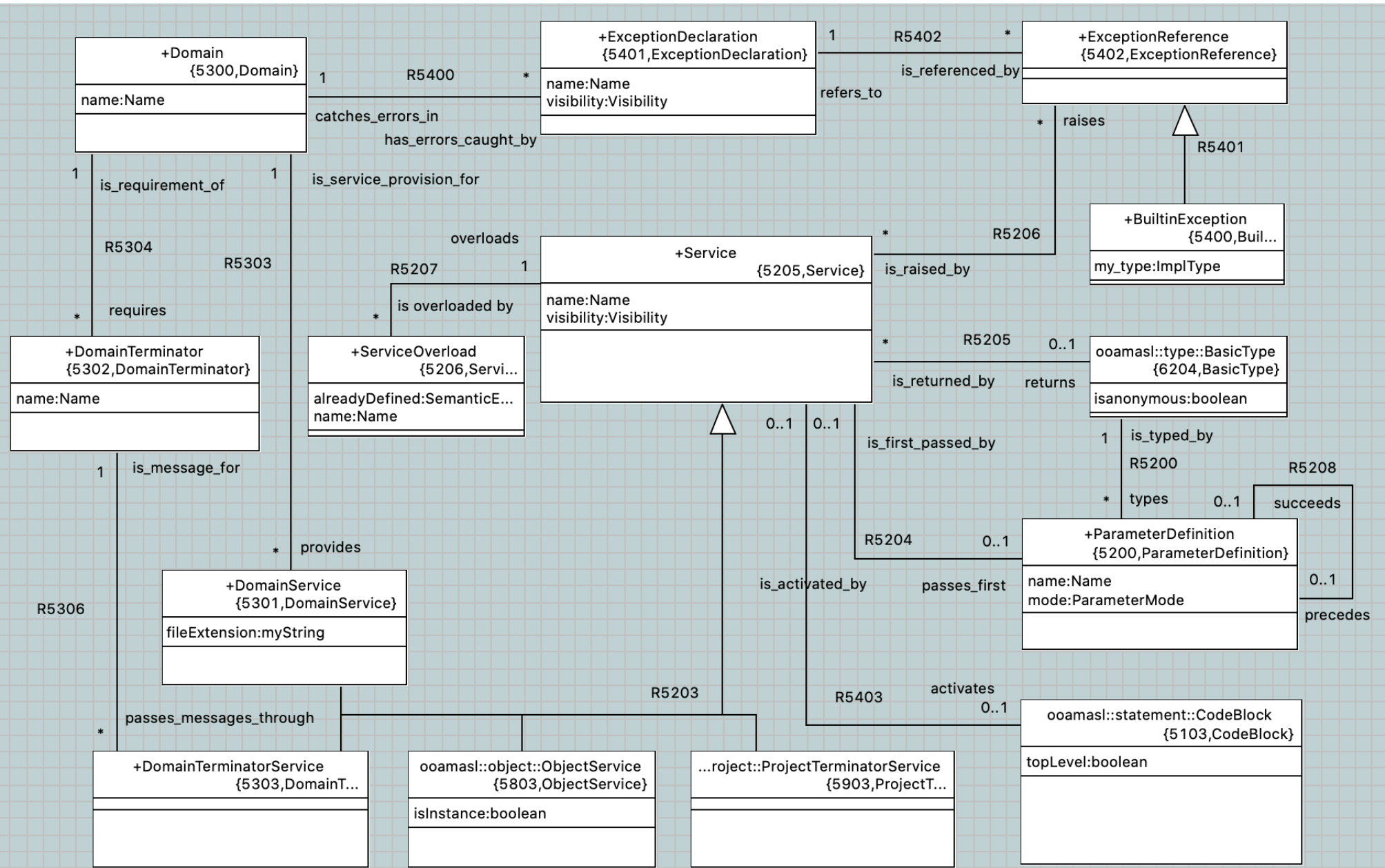




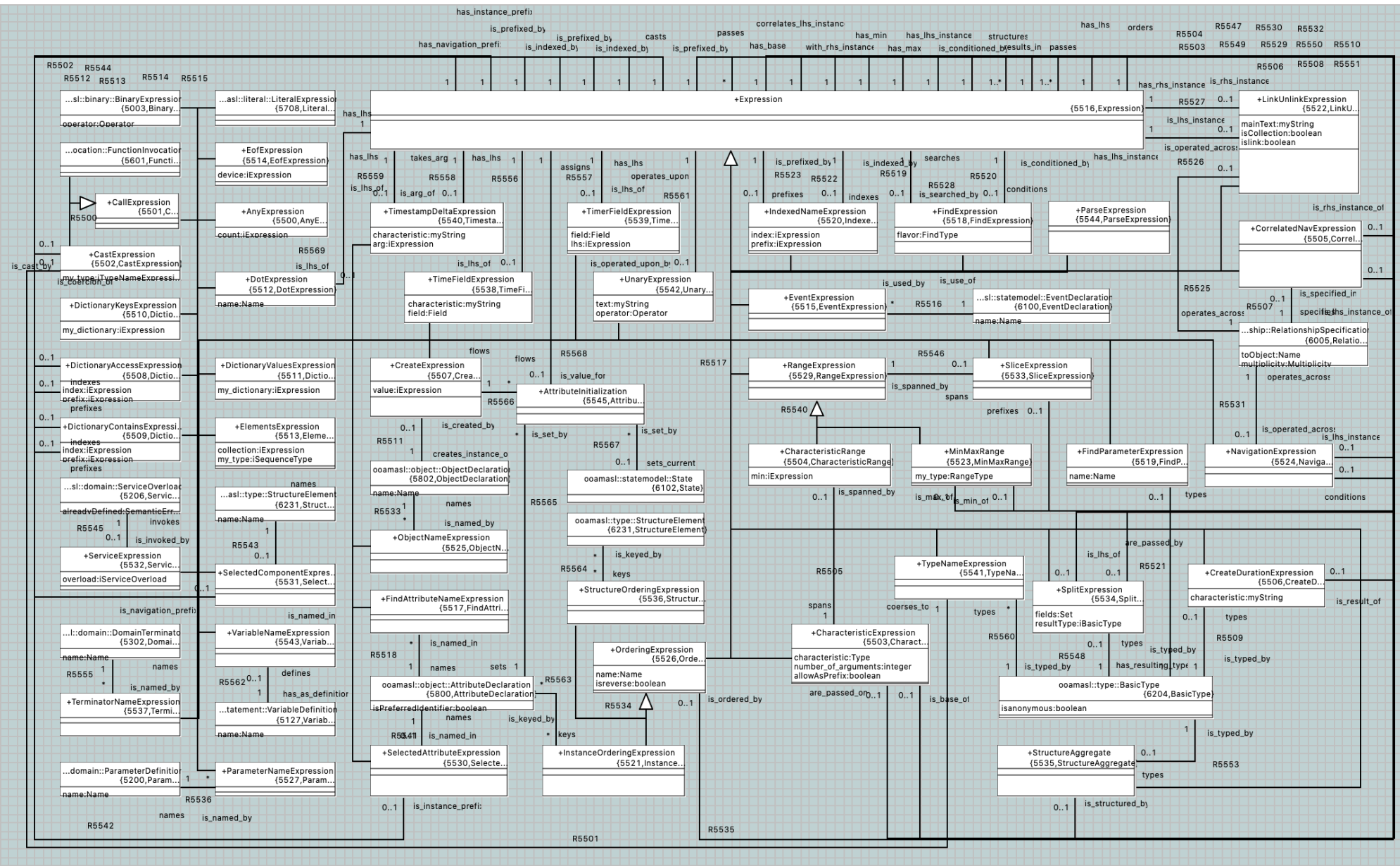
# OOA of MASL: builtin



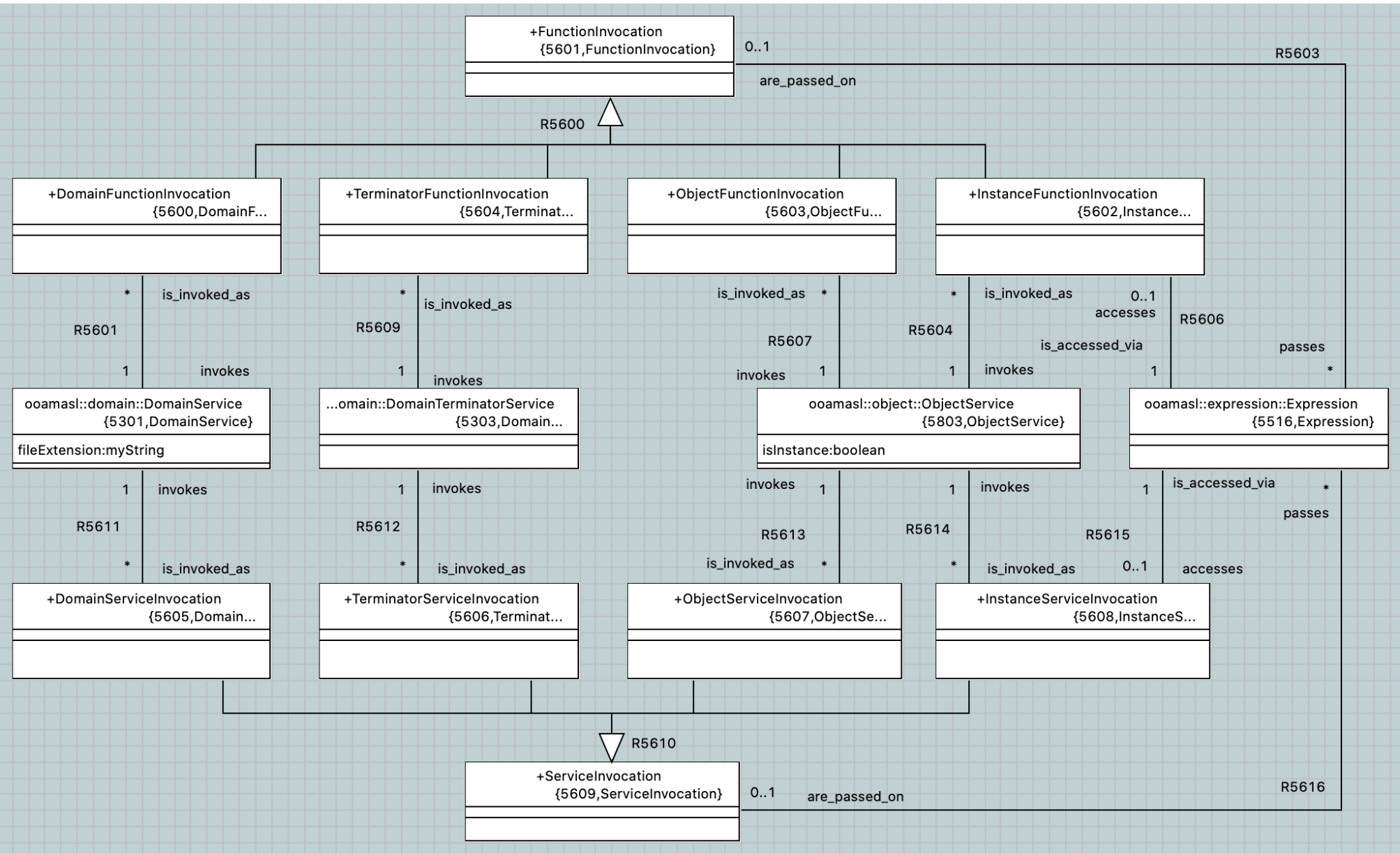
# OOA of MASL: domain



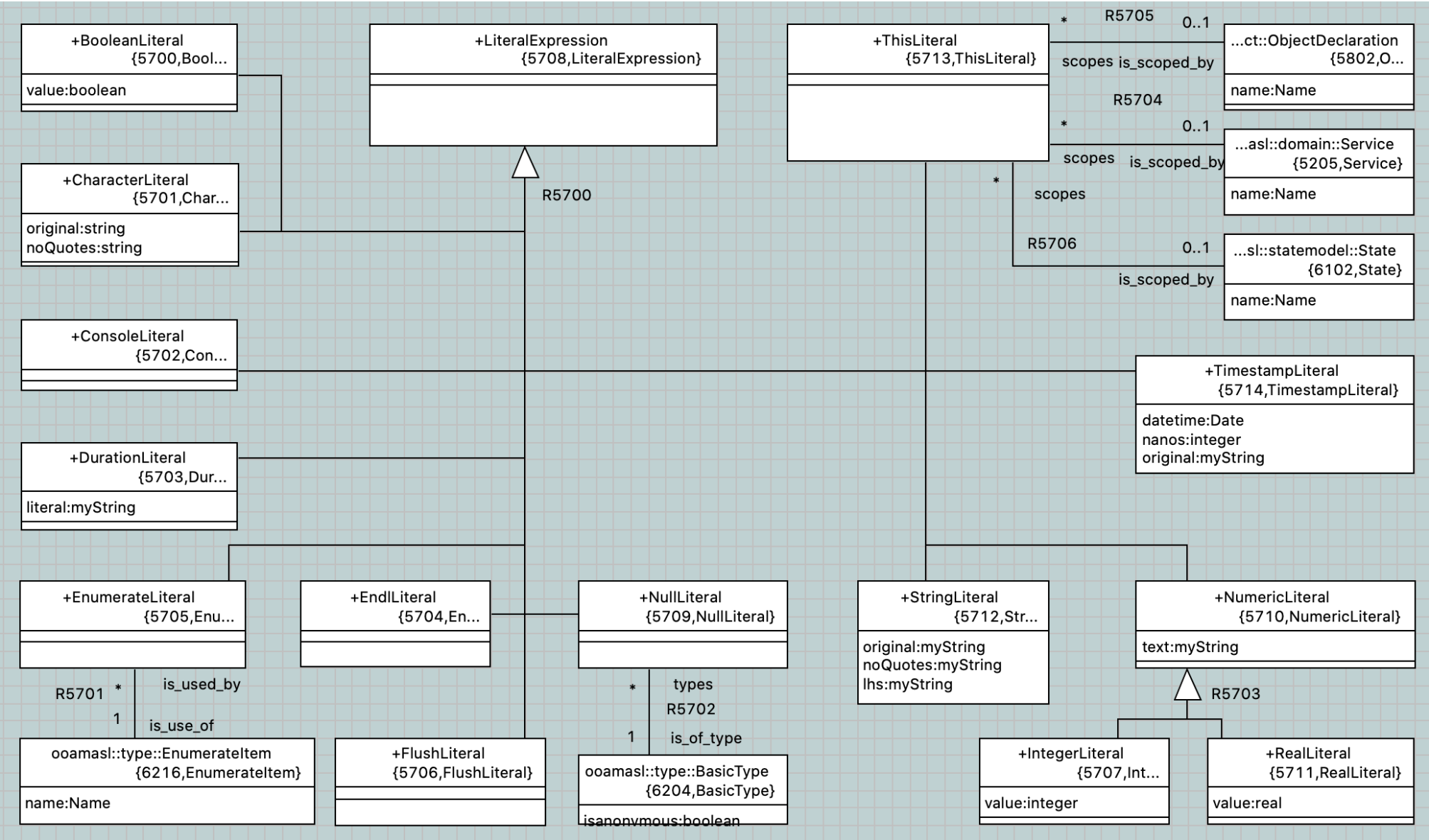
# OOA of MASL: expression



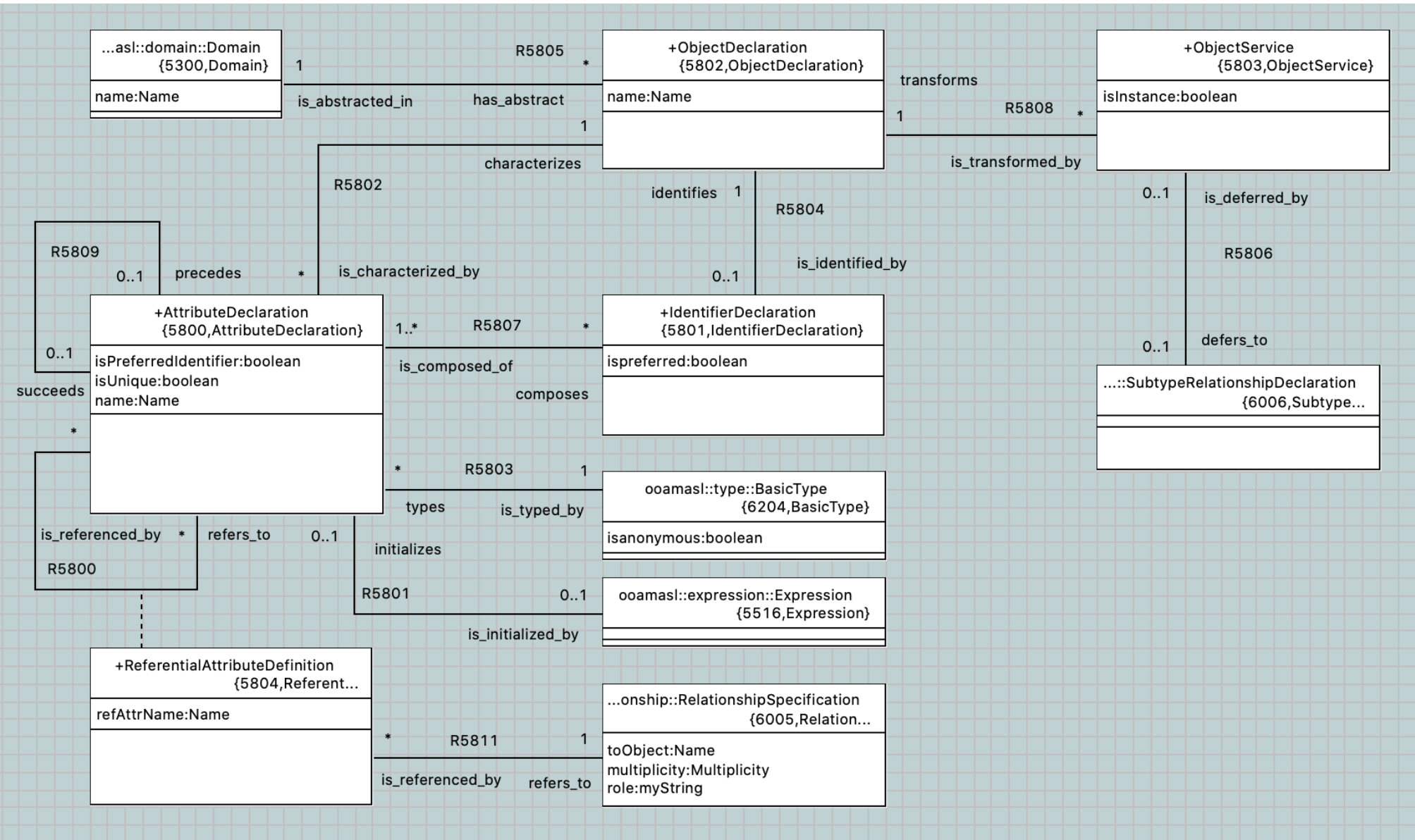
# OOA of MASL: invocation



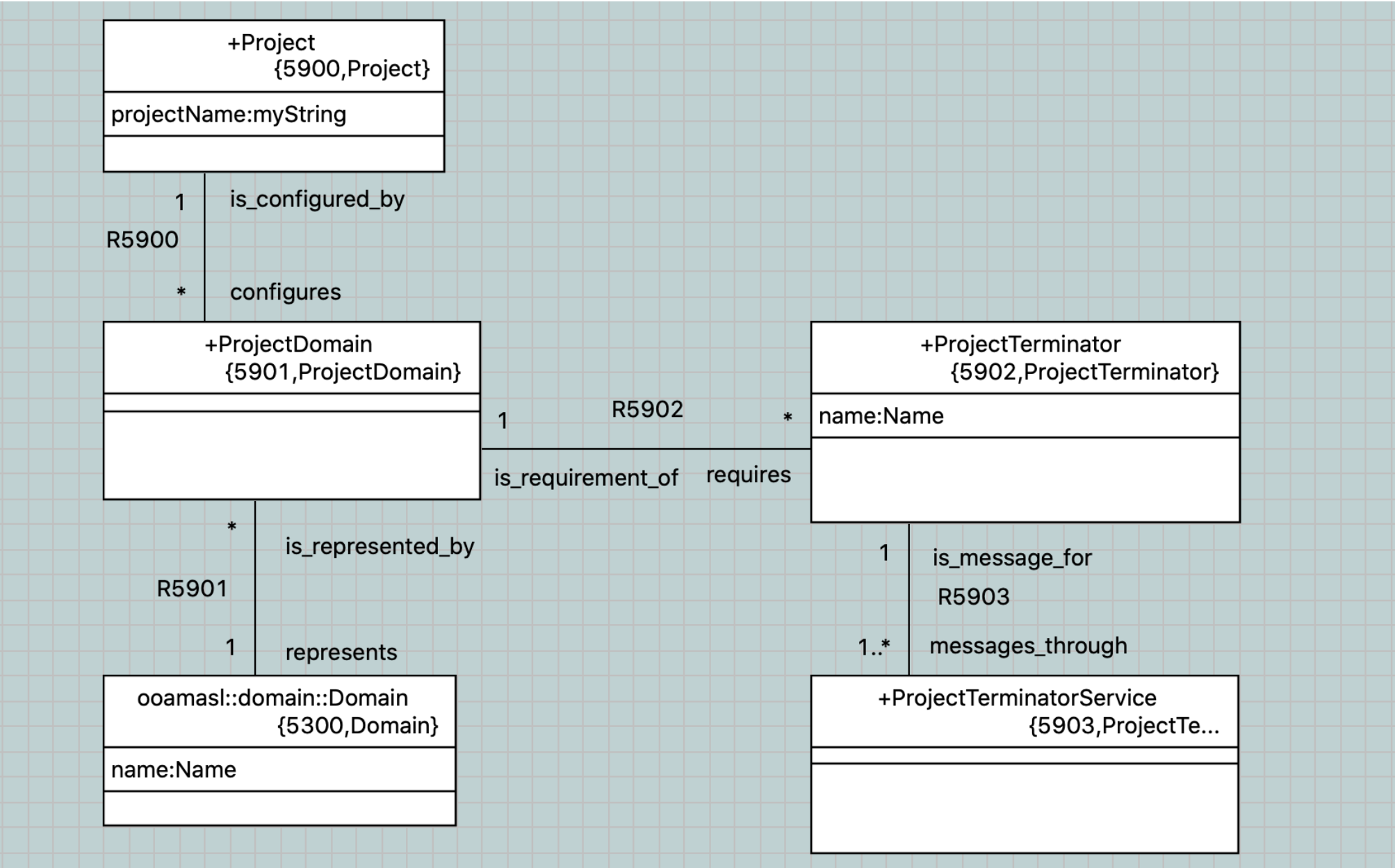
# OOA of MASL: literal



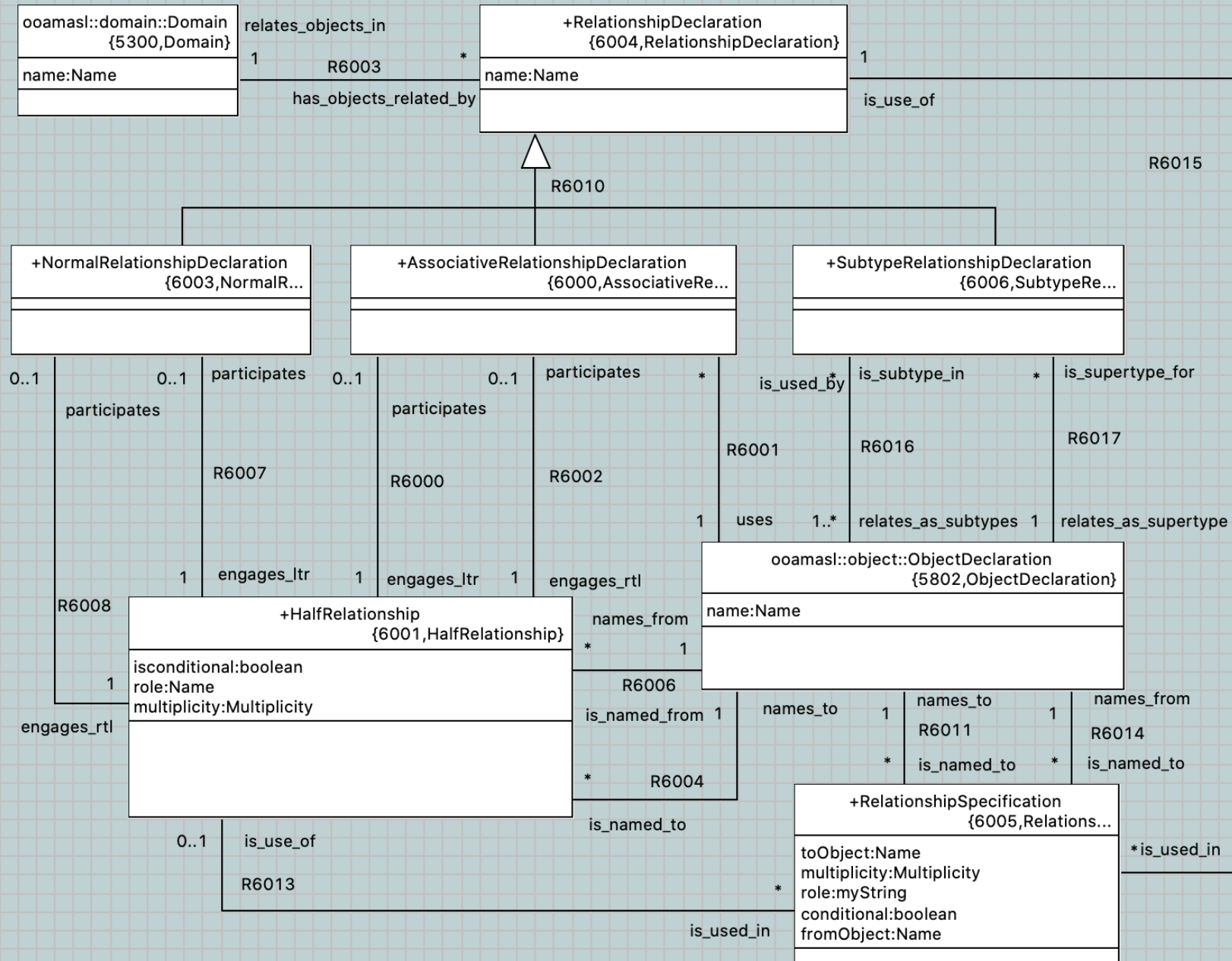
# OOA of MASL: object



# OOA of MASL: project

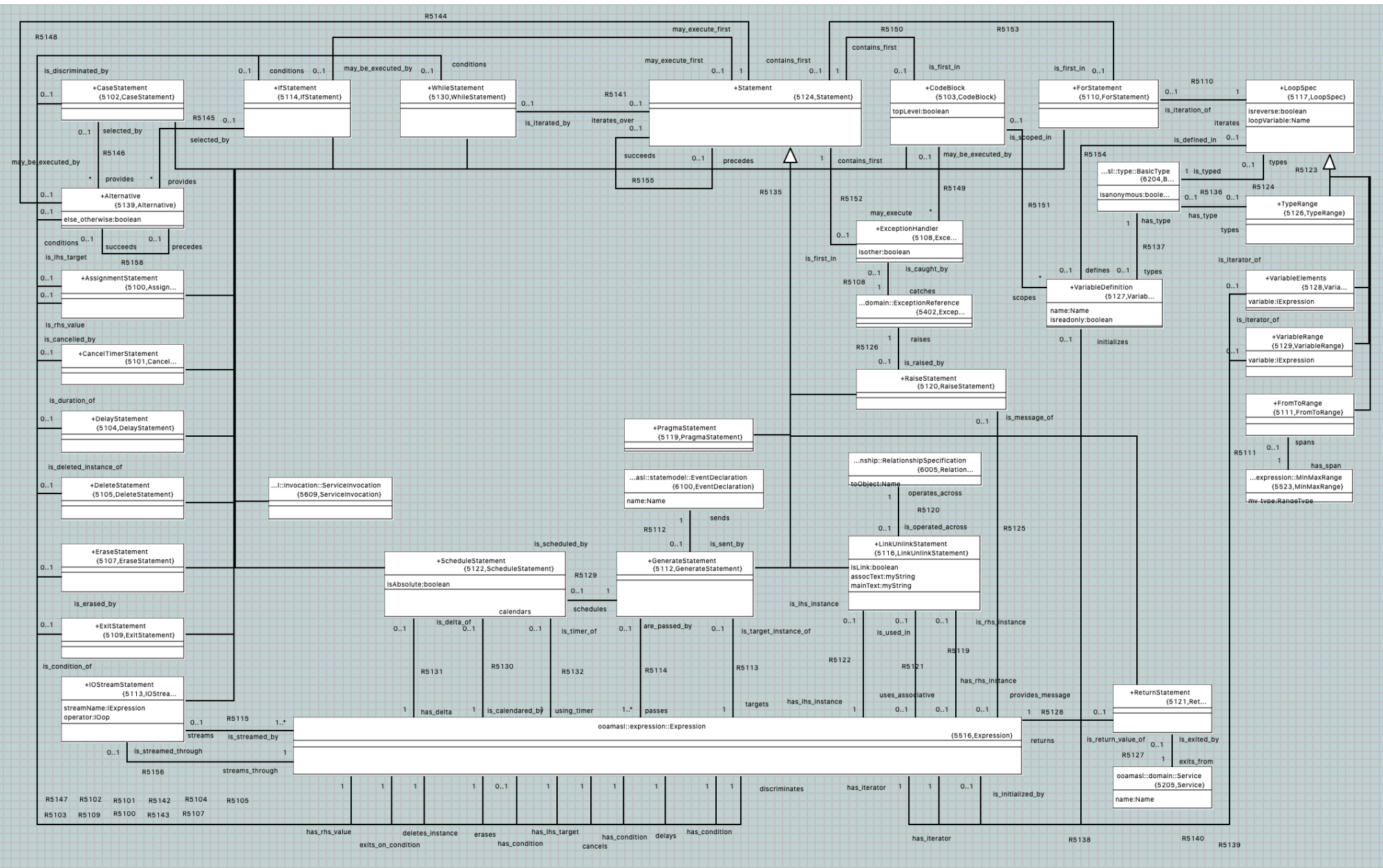


# OOA of MASL: relationship

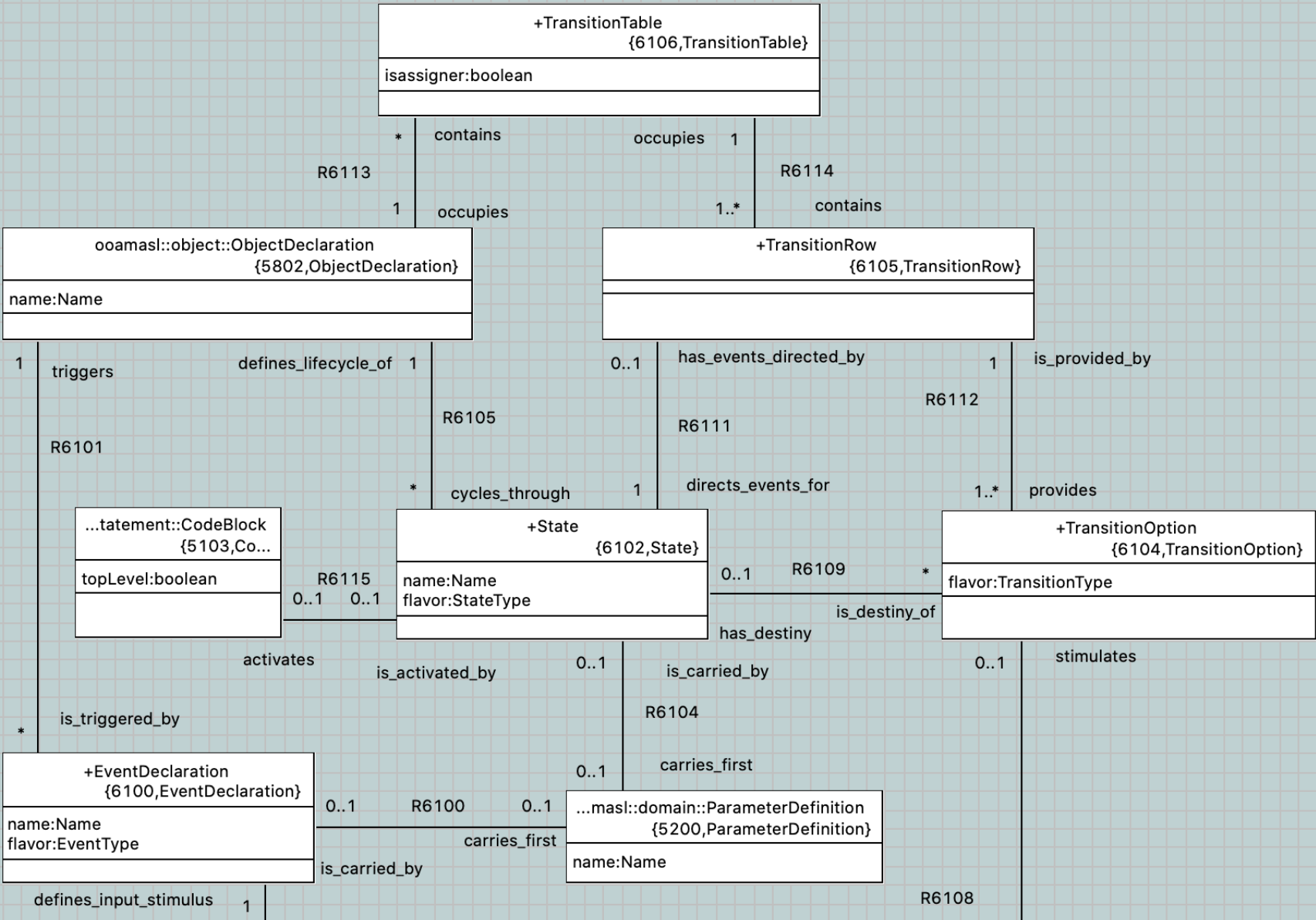




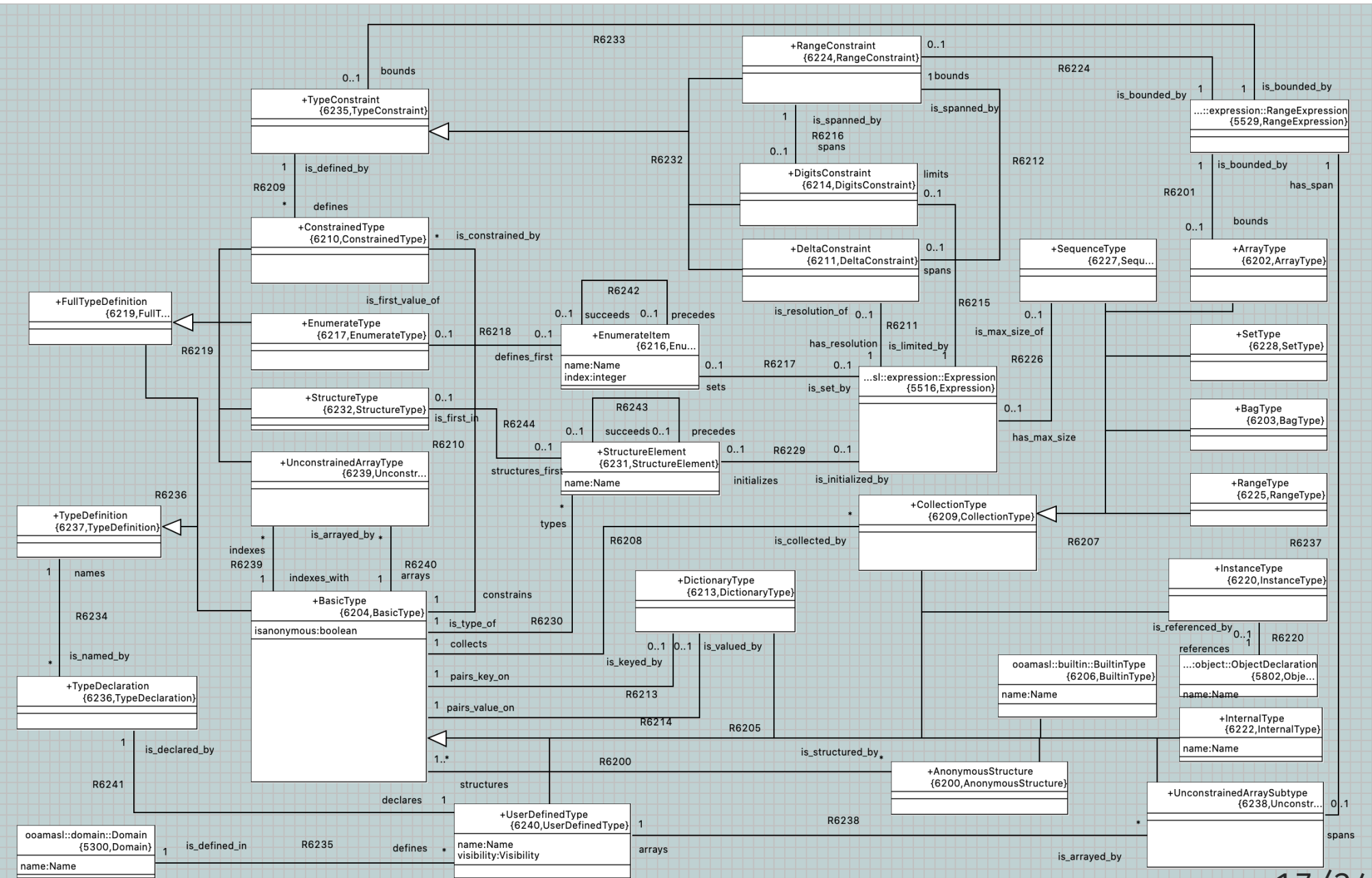
# OOA of MASL: statement



# OOA of MASL: statemodel

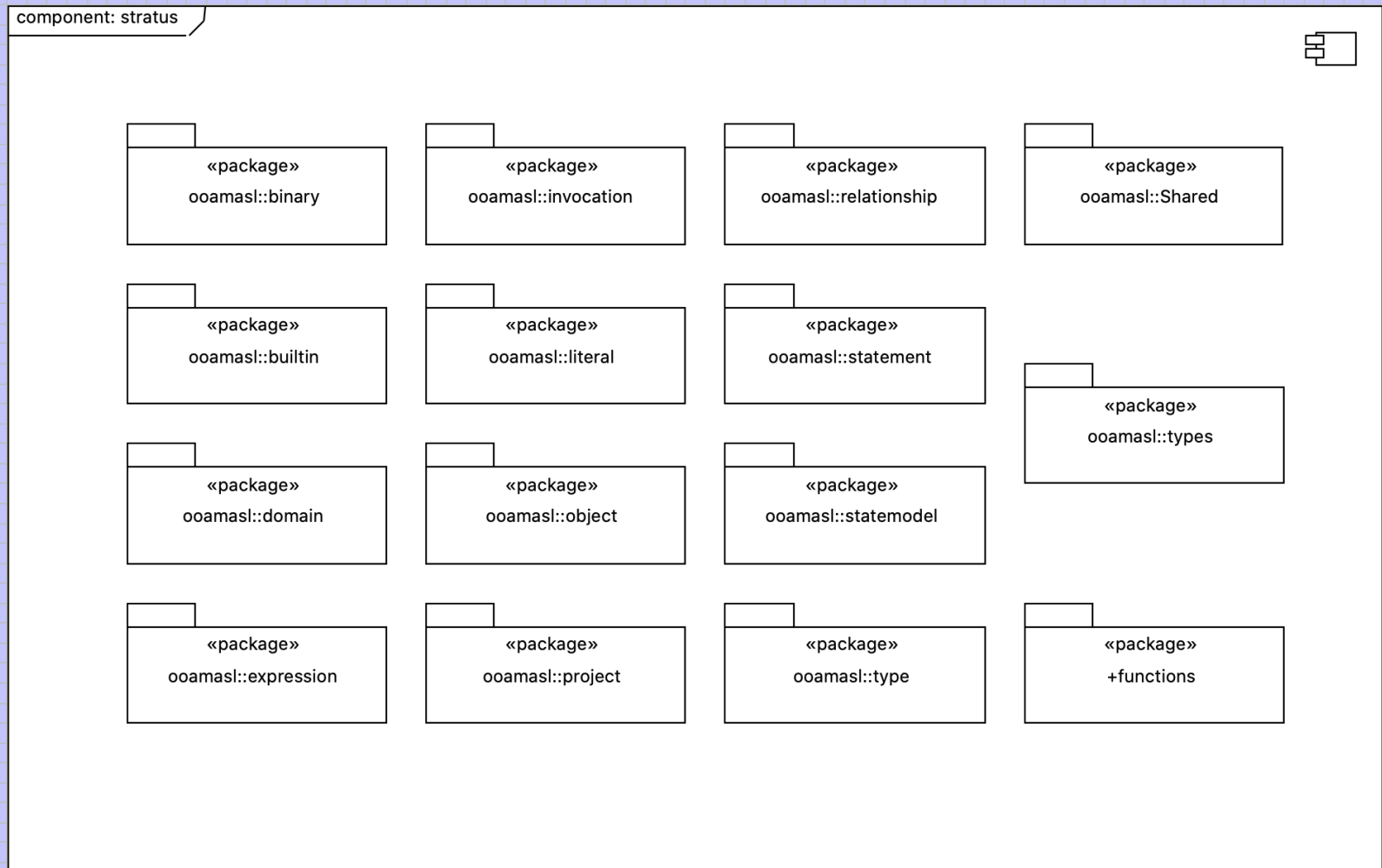


# OOA of MASL: type



# OOA of MASL: Stratus Deployment

```
▶ ooamasl [mc 11745_loadmasl3]
▶ runtime [ciera master]
▼ > stratus [mc 11745_loadmasl3]
  ▼ deploy
    ▼ +stratus
      ooamasl::binary
      ooamasl::builtin
      ooamasl::domain
      ooamasl::expression
      ▼ functions
        builtin_types
        ▶ create_RelationshipSpe
        init
        render_HelloWorld
        render_metamodel
        ▶ resolve_name
        ▶ select_any_AttributeDe
        ▶ select_any_BasicType_v
        ▶ select_any_DomainServ
        ▶ select_any_DomainTerm
        ▶ select_any_EventDeclar
        ▶ select_any_ExceptionDe
        ▶ select_any_ObjectDecla
        ▶ select_any_ObjectServi
        ▶ select_any_ProjectTerm
        ▶ select_any_Relationship
        ▶ select_any_State_relate
        ▶ select_any_State_where
        ▶ select_any_TypeDeclar
        ▶ select_any_UserDefined
        ▶ select_any_VariableDefi
      ooamasl::invocation
      ooamasl::literal
      ooamasl::object
      ooamasl::project
      ooamasl::relationship
      ooamasl::Shared
```



# package references and *naked* OOA of MASL

- A package reference is a package linked to another package in another location (package or project).
  - Package references allow reuse and *import* of subsystems.
- The term *naked OOA* refers to a class model with no state or functional activity intended to be used as the schema for multiple projects.
  - A meta-model is most useful as a naked class model.

# Statistics

## *Statistics*

subsystems	12
classes	176
attributes	117
relationships	230
production rules	166
generated LOC (;)	51292

# MASL loader

- deployment model
  - imported OOA of MASL subsystem packages
  - loader helper functions (OAL)
- parser as `antlr` lexer / parser
- loader as `antlr` walker with invocations into deployment model
- model API functions
  - Ciera provides an API which allows `create`, `relate`, `set_attribute` and `call_function`

# MASL loader

- action language functions
- LOAD class interface to generated model Java
- Ciera provides a build chain based upon maven.
- Ciera generated code
- calculator ALU has been primary test model.
- Stratus can parse and load Stratus.
- Generate Hello, world.
- Generate abstract classes with stratus of stratus.
- Generate inventory report of major elements sorted in some order.



# MASL loader: key functions

- init
- Builtin/InternalType\_populate
- select\_any\_ObjectDeclaration\_where\_name
- resolve\_name
- create\_RelationshipSpecification
- ReferentialAttribute\_resolve

# next steps

- review, further test and refinement
- template and rendering HOWTO
- cloud template integration
- Java translation of MASL activities