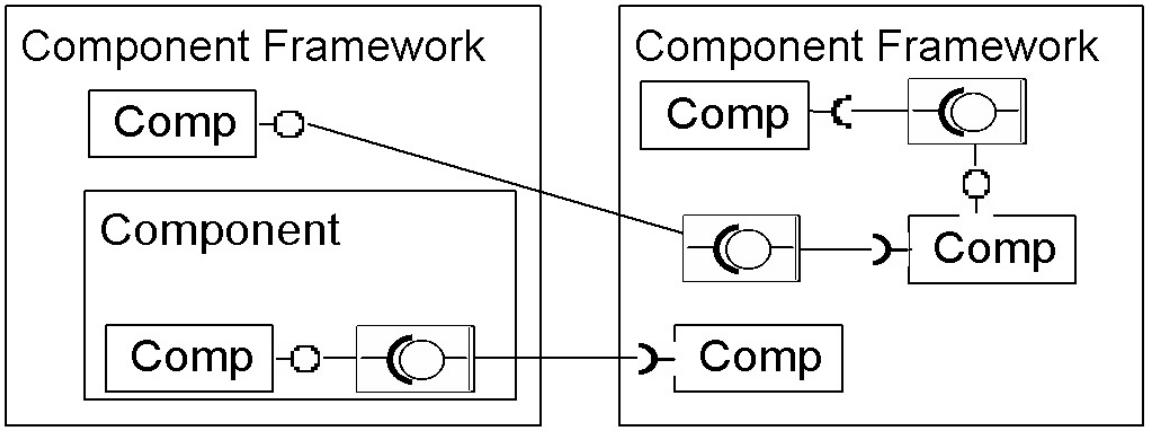
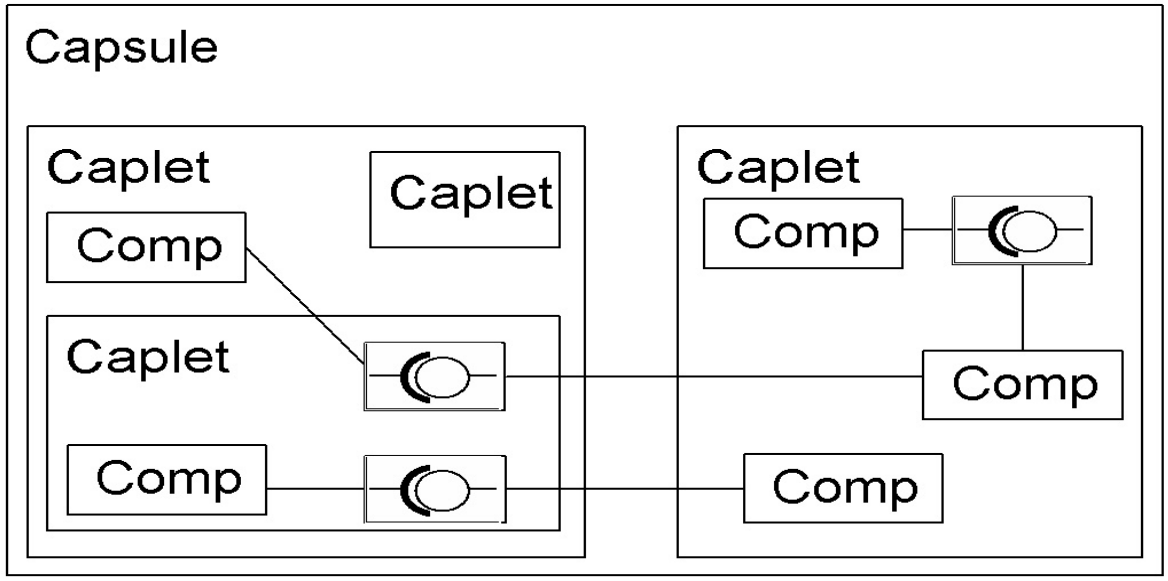
Dynamic Component System



Design Time Semantics

Component – Functionality Owner
Interface – Interaction Point Owner
Receptacle – Interaction Point Owner
Binding – Communication Owner
Component Framework – Constraint Owner

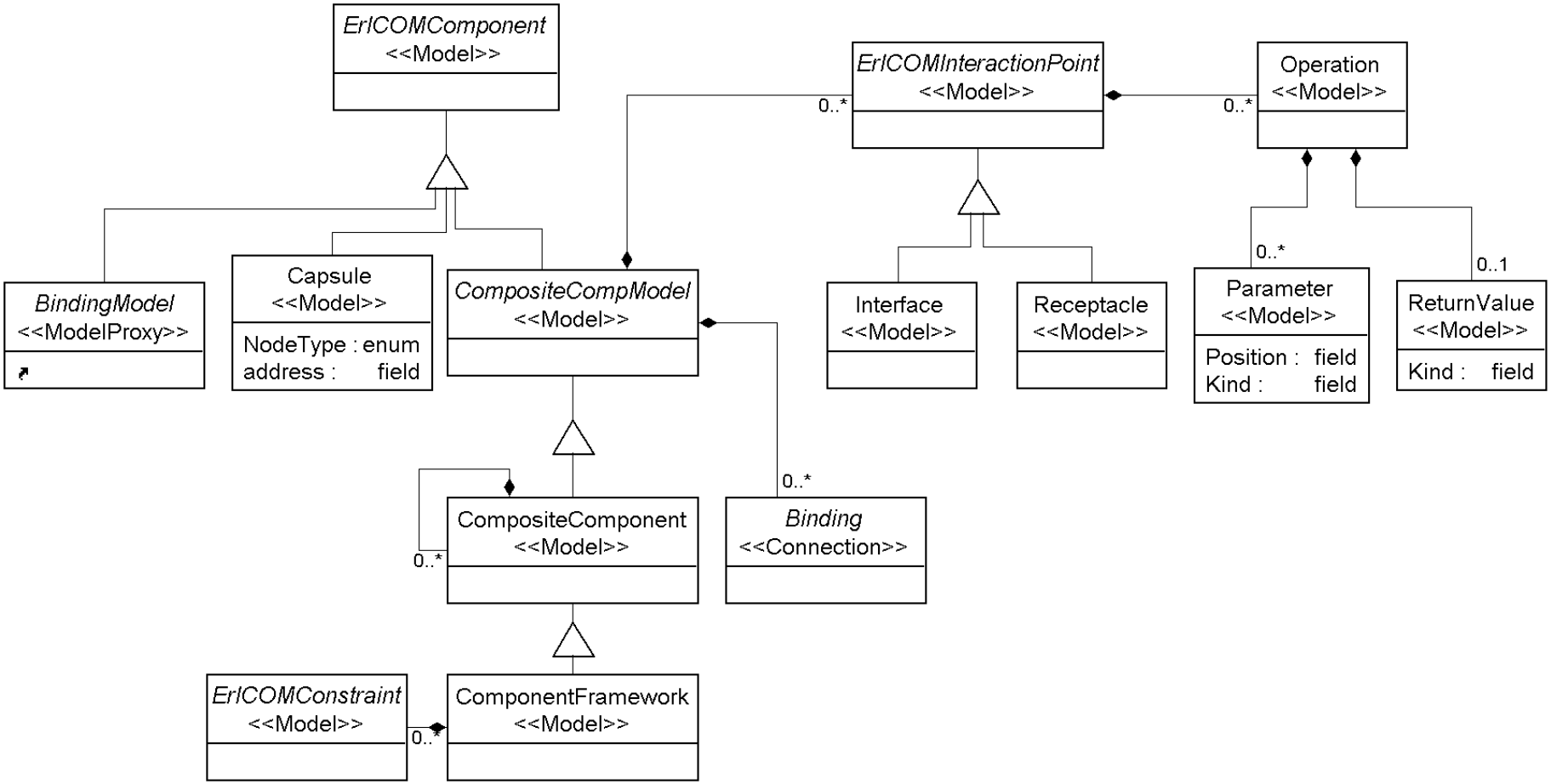


Deployment Configuration

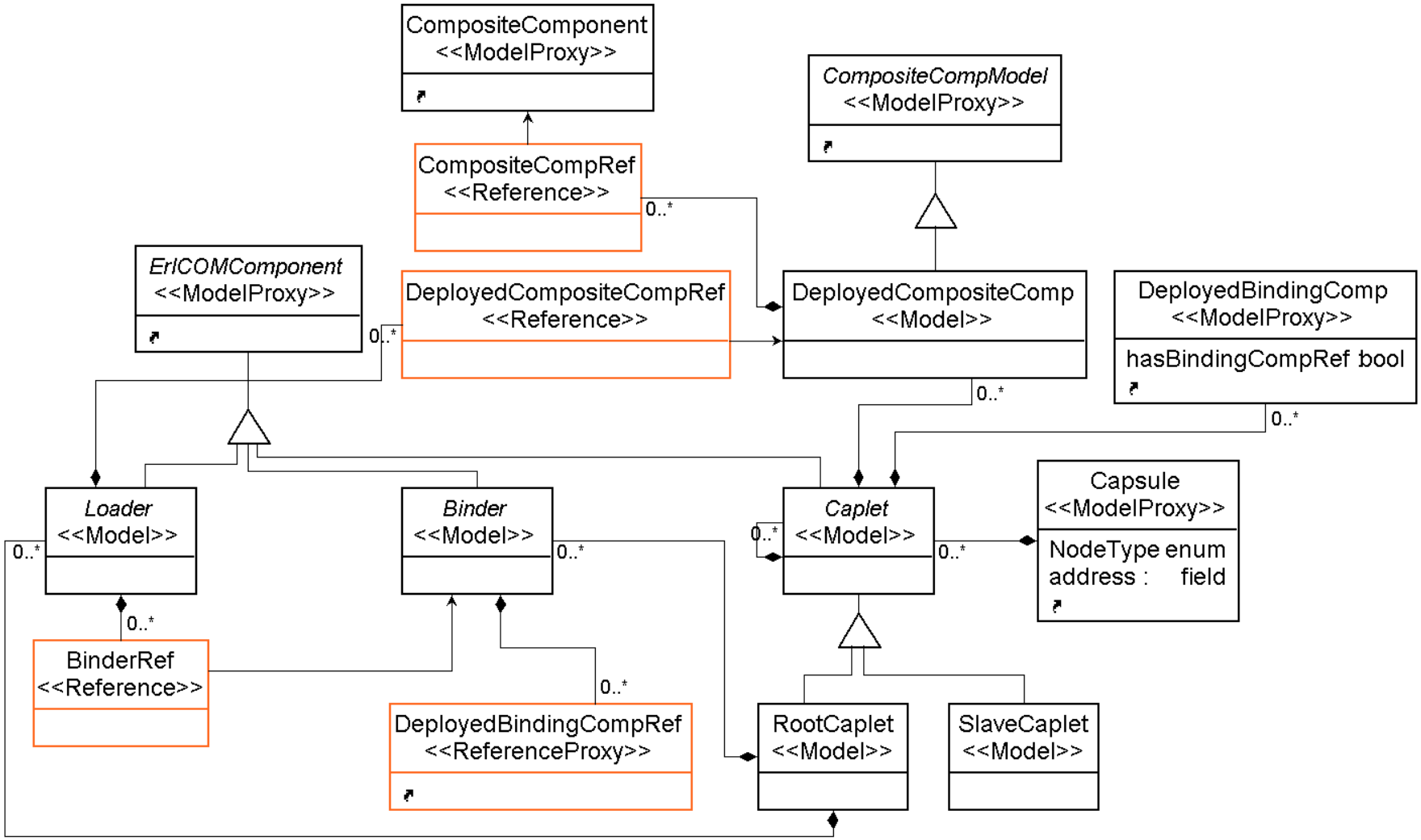
Capsule – Supervision Owner
Caplet – Component Owner

ErlCOM is a reference implementation for IST FP6 RUNES demo

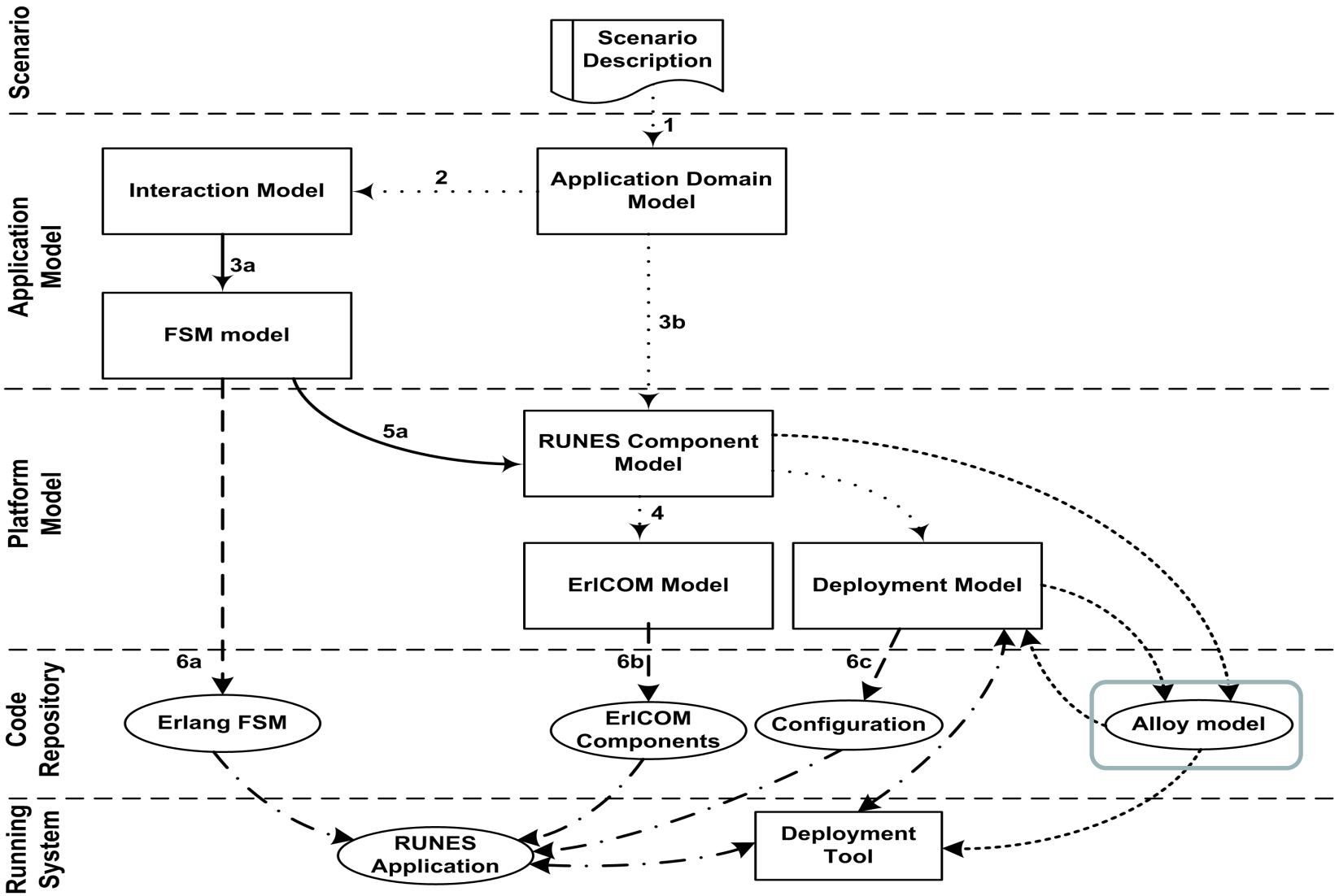
ErlCOM Kernel Metamodel



ErlCOM Deployment Metamodel



Development Process



Alloy & Alloy Analyzer

evopro

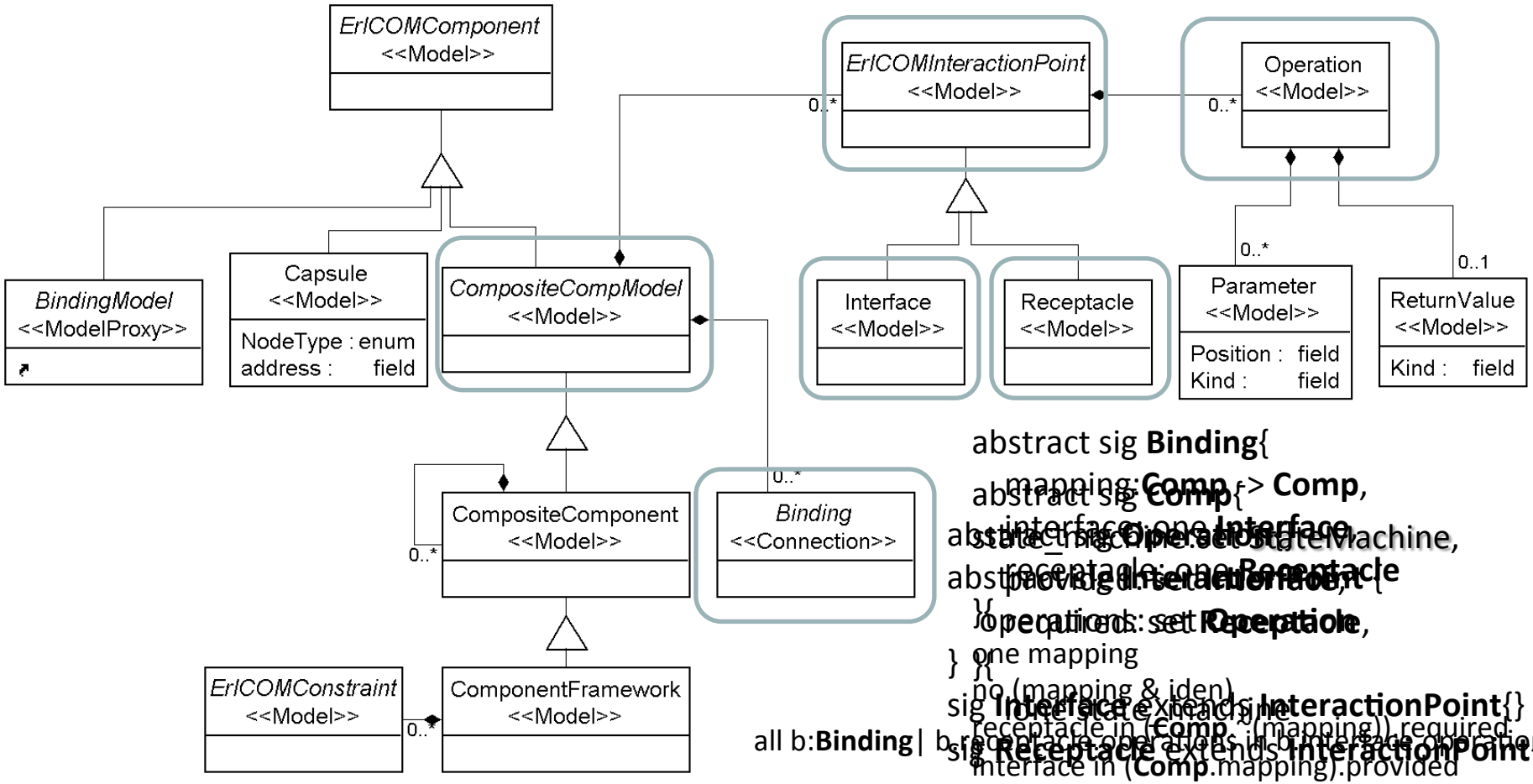
GOOD things

- Provides formal specification instead of mere "wishful thinking"
- Facilitates software abstraction via precise and unambiguous notation
- Translates specifications into Boolean expression to be automatically analyzed by SAT solvers (of your choice)
- Encourages incremental analysis of problem domain via successive solution specifications
- Produces both examples and counterexamples for predicates
- Alloy Analyzer automates finding models that satisfy specification

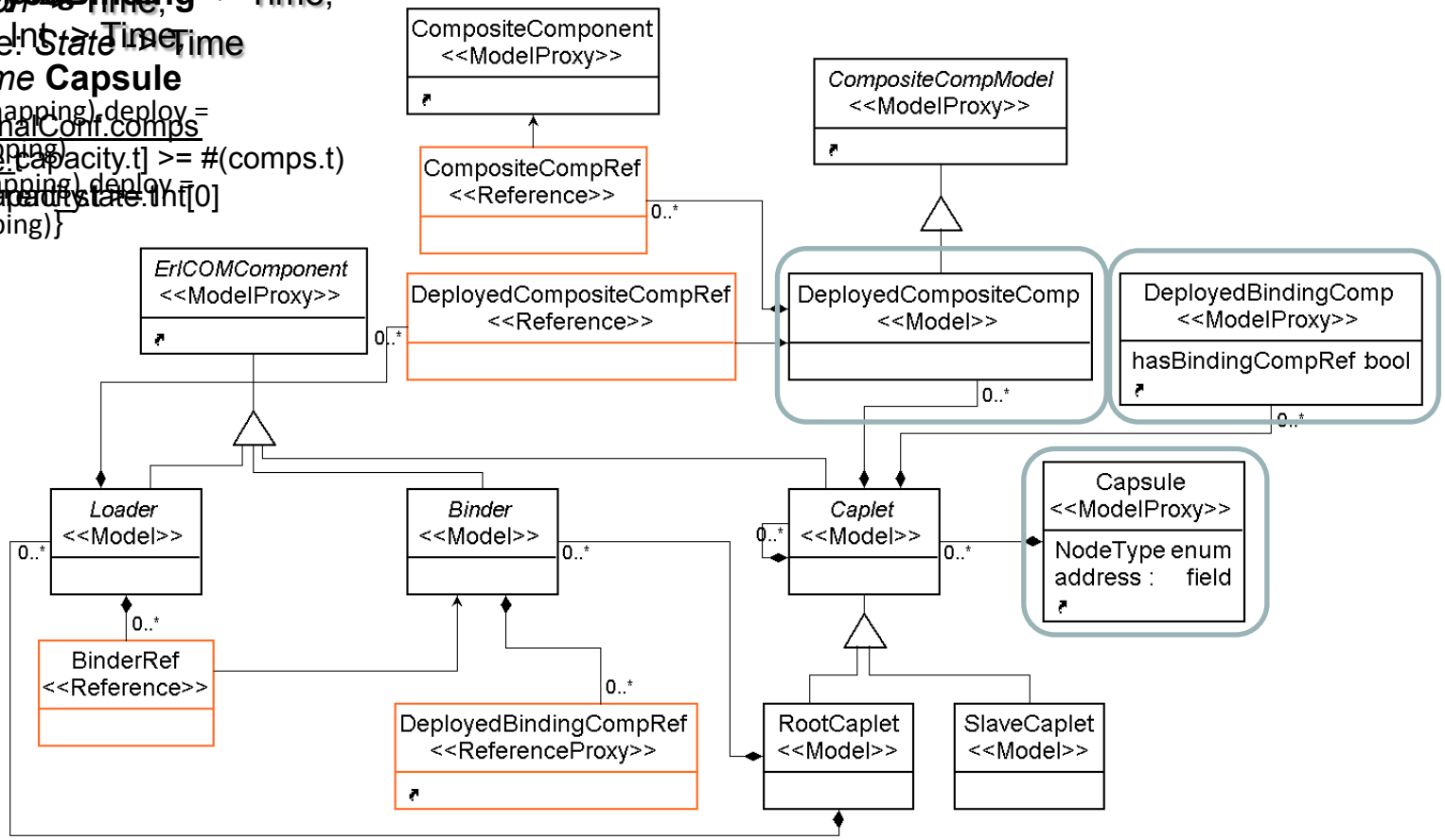
BUT

- **Never** allows you to prove that your model is **correct**
- **Only** attempts to find counterexamples **within a limited scope** that violate the constraints of specified system

Decorated Kernel Metamodel



Decorated Deployment Metamodel



Additional Semantics in Alloy

```
abstract sig State{}
abstract sig Transition{
   trans: State -> State
}{
one trans
}
abstract sig StartState extends State{}
abstract sig StartTransition extends Transition{}
pred transition[d:DeployedComp,t,t':Time]{
   (d.fire.t).trans.State = d.current_state.t
   (d.fire.t).trans[State] = d.current_state.t'
}
abstract sig StateMachine{
states: some State,
startState: one StartState,
transitions: some Transition,
startTransition: one StartTransition,
}{
no (states & startState)
no (transitions & startTransition)
}
fact Traces{
...
all t:Time-TO/last[],d:DeployedComp|let t'=TO/next[t]|
some d.fire.t => (transition[d,t,t'])
all t:Time|some DeployedComp.fire.t
}
```

Finite State Machine

Component Migration

```
pred migrate(c_src,c_dst:Capsule,d:DeployedComp,t,t':Time){
c_src != c_dst
#(c_dst.comps.t) < int[c_dst.comp_capacity.t]
c_dst.comps.t' = c_dst.comps.t+d
c_src.comps.t' = c_src.comps.t-d
all capsule:Capsule|capsule.bindings.t'=capsule.bindings.t
all capsule:Capsule-c_src-c_dst| capsule.comps.t'=capsule.comps.t
all capsule:Capsule-c_src-c_dst| capsule.comp_capacity.t' = capsule.comp_capacity.t
}
```

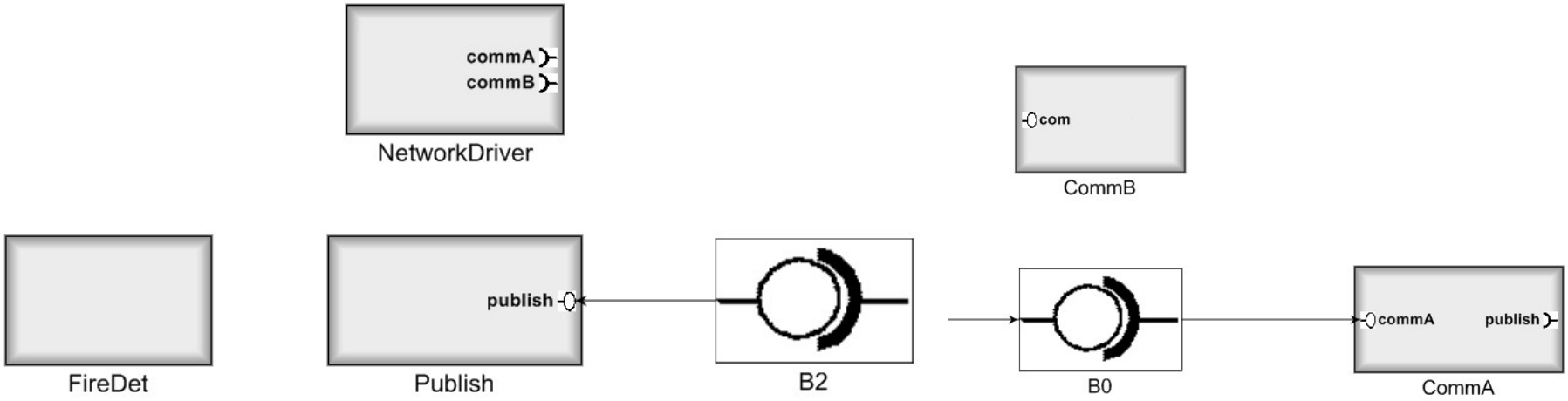
```
no deployedBinding:DeployedBinding|some t:Time|
deployedBinding in Capsule.bindings.t and
(deployedBinding.mapping.DeployedComp not in Capsule.comps.t
or deployedBinding.mapping[DeployedComp] not in Capsule.comps.t)
```

Additional Constraints

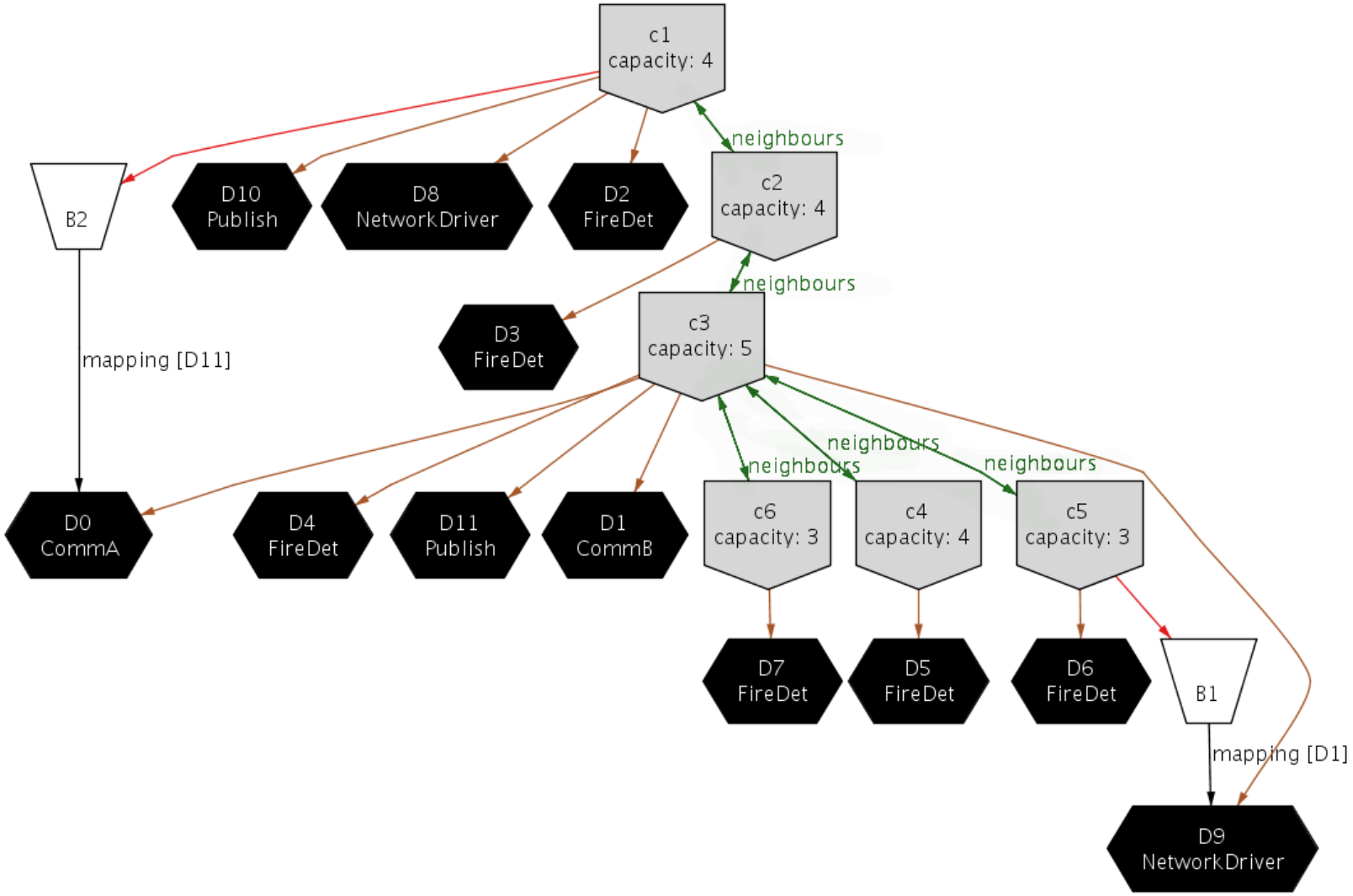
```
no disj capsule1,capsule2:Capsule|
some (capsule1.bindings) & (capsule2.bindings)
no disj capsule1,capsule2:Capsule|
some (capsule1.comps) & (capsule2.comps)
```

```
no disj b1, b2:DeployedBinding| (b1.deploy = b2.deploy)
and (b1.mapping.DeployedComp = b2.mapping.DeployedComp)
```

Model in Modeling Tool



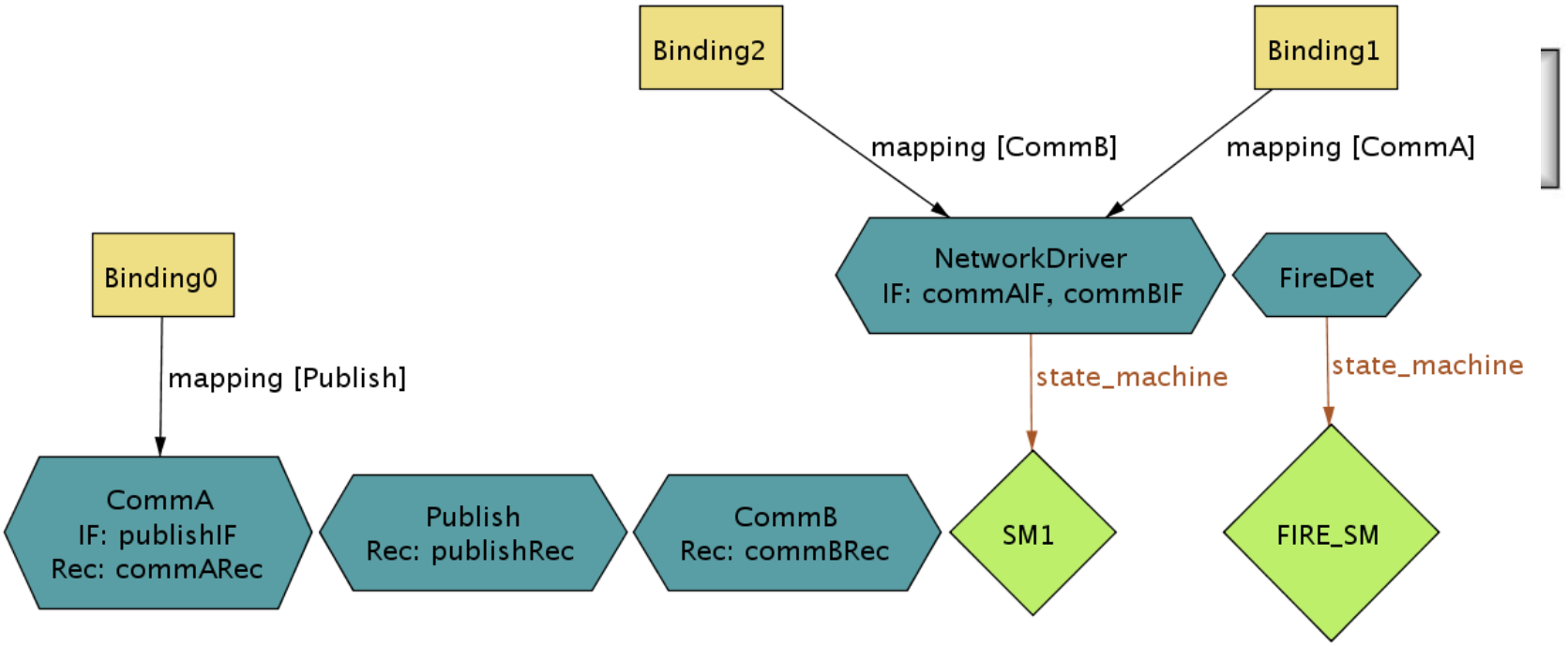
Model in Alloy Analyzer



Simplified Fire Scenario

- Two capacity limited **capsule**s represent deployed „sensor network”
- Communication enabling **component**s are deployed within capsules
- **Fire Detector** component controls capsule’s capacity dynamics
- Capsules’ capacity constraints enforce component migration
- **Network Driver** autonomously changes communication port

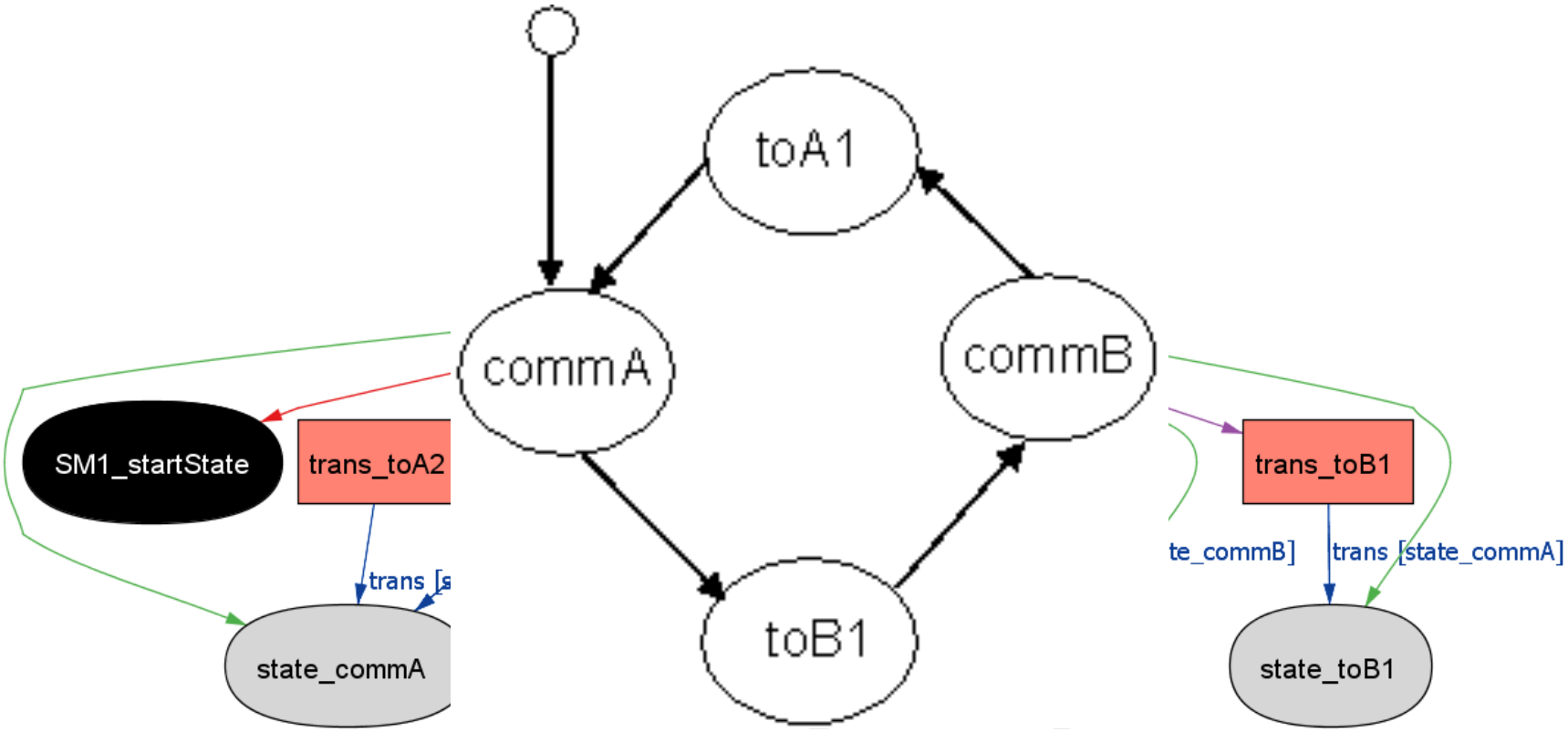
Functional Configuration



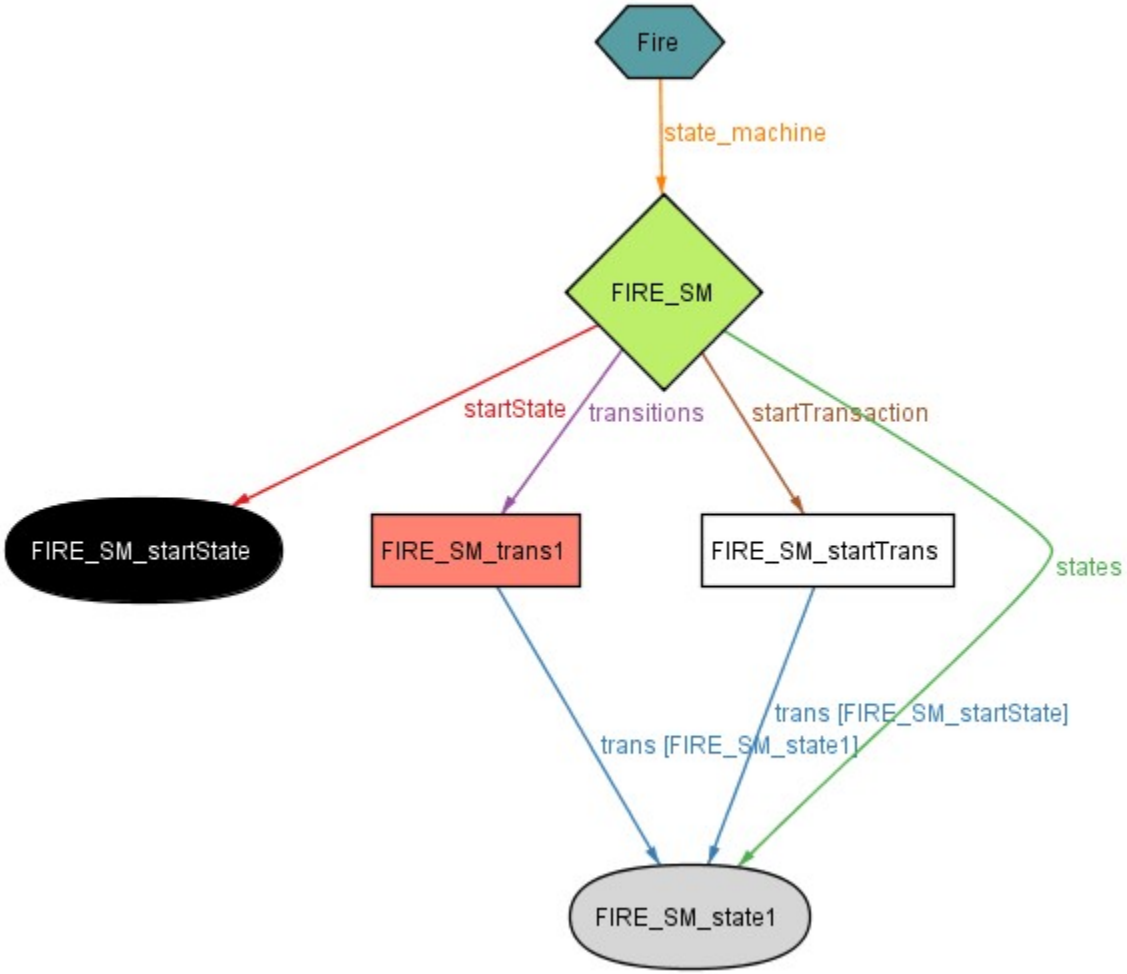

```
fact{
        FunctionalConf.comps == CommA+CommB+NetworkDriver+Publish+Fire

        #(FunctionalConf.comps <: CommA)==1
        #(FunctionalConf.comps <: CommB)==1
        #(FunctionalConf.comps <: NetworkDriver)==1
        #(FunctionalConf.comps <: Publish)==1
        #(FunctionalConf.comps <: Fire)==1
        ...
}
```

FireDet

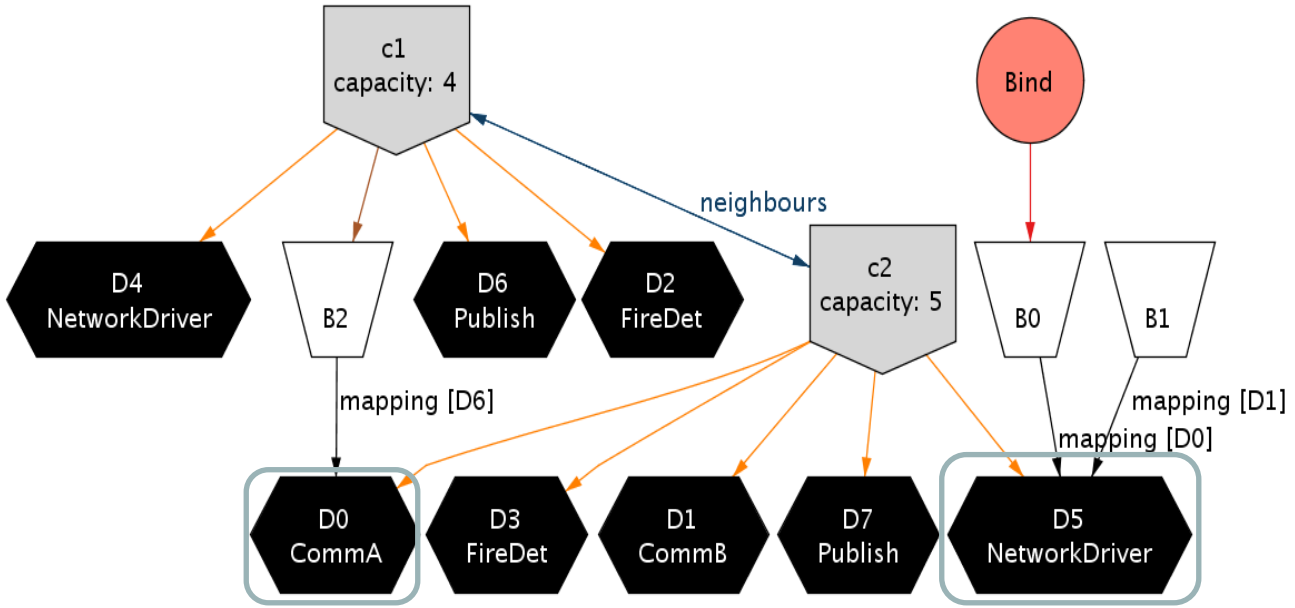
NetworkDriver's State Machine



FireDetector's State Machine

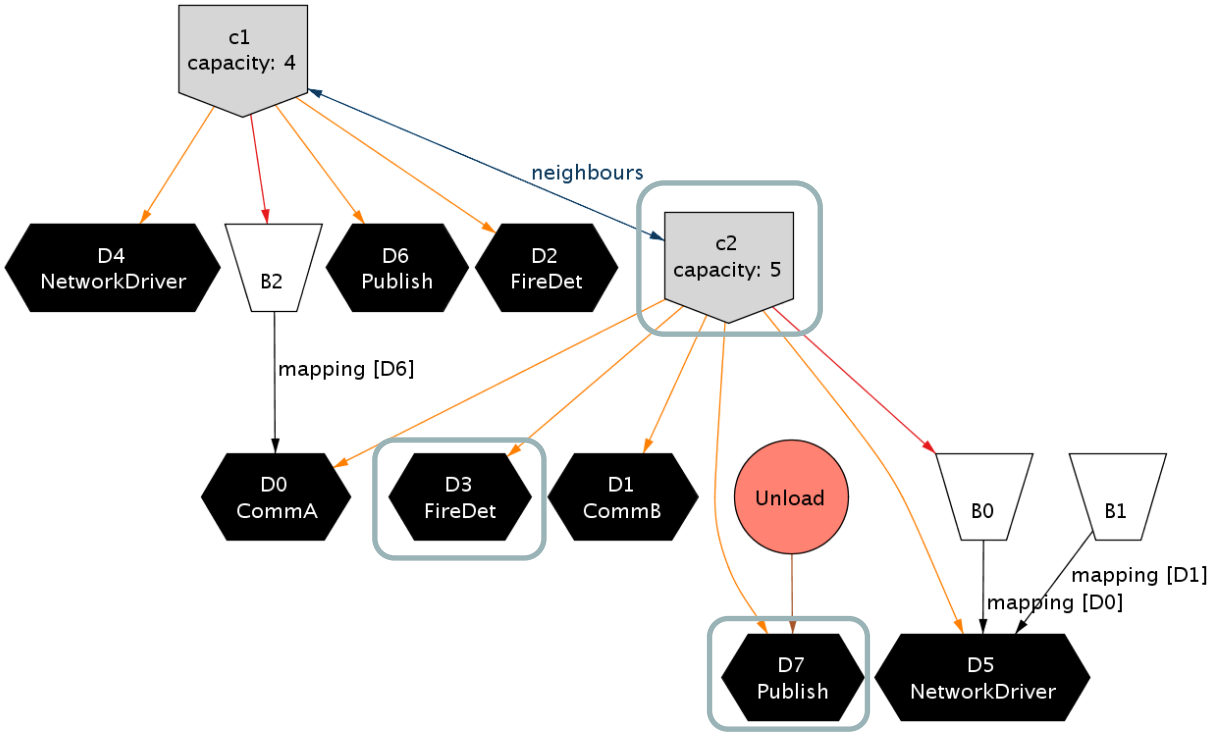


Alloy Trace Sequence No1



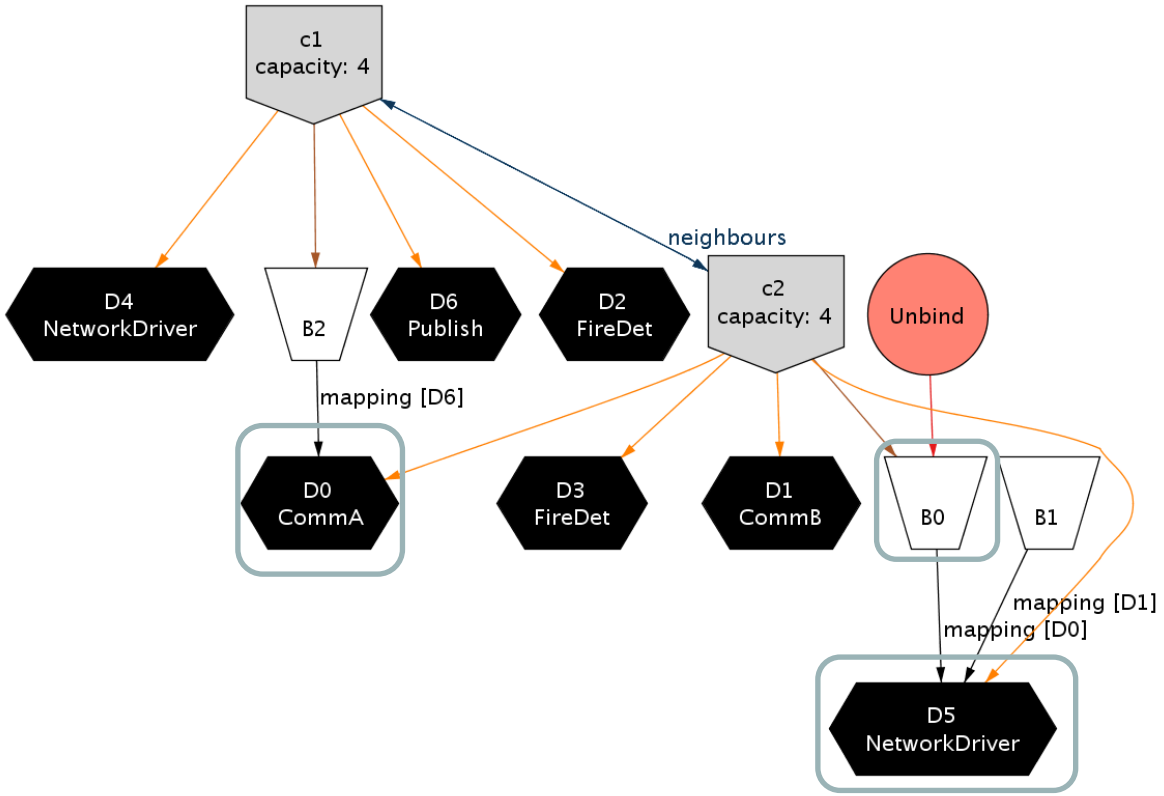
- **SM1_startTrans** is activated, its action part initiates a new binding session
- Binding session (**Bind**) creates deployed **binding (B0)**
- **Network Driver (D5)** is to be bound to **CommA (D0)**

Alloy Trace Sequence No2



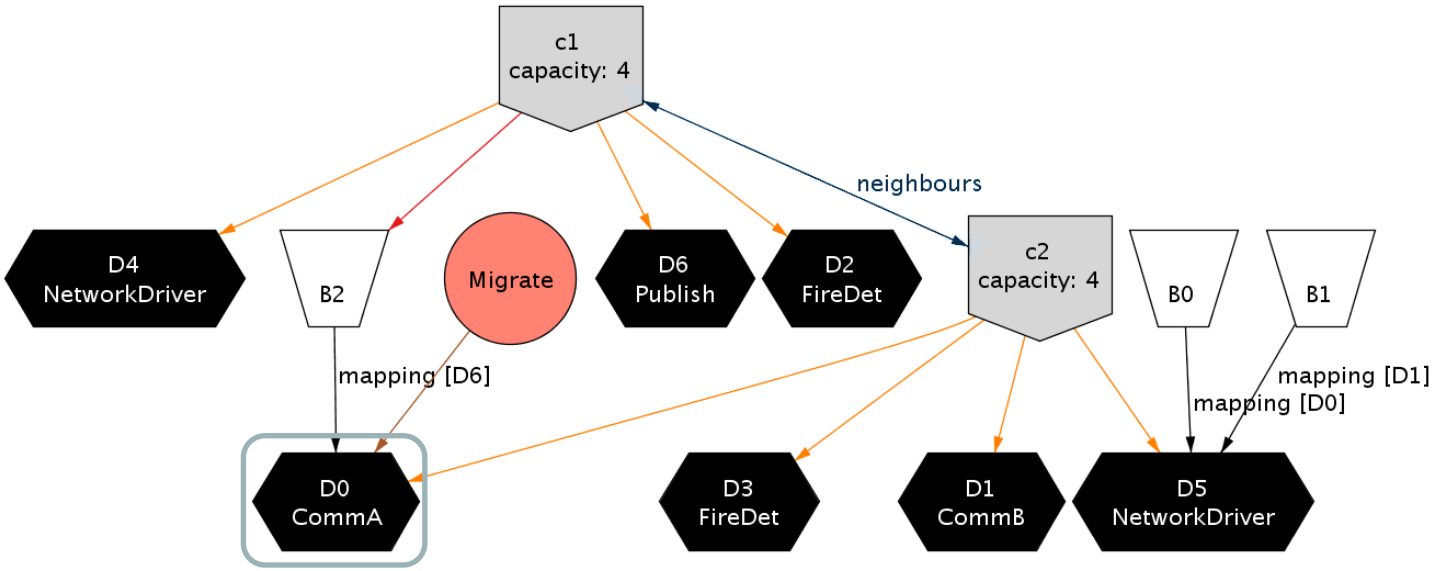
- **D3**'s state machine (**FIRE_SM**) gets activated, it will reduce **c2**'s capacity
- **c2**'s capacity limit enforces component migration from **c2**
- **c1** also has **Publish** already deployed, so migration is demoted to **Unload**
- **D7** will get unloaded, **c2**'s capacity limit is to be set to 4

Alloy Trace Sequence No3



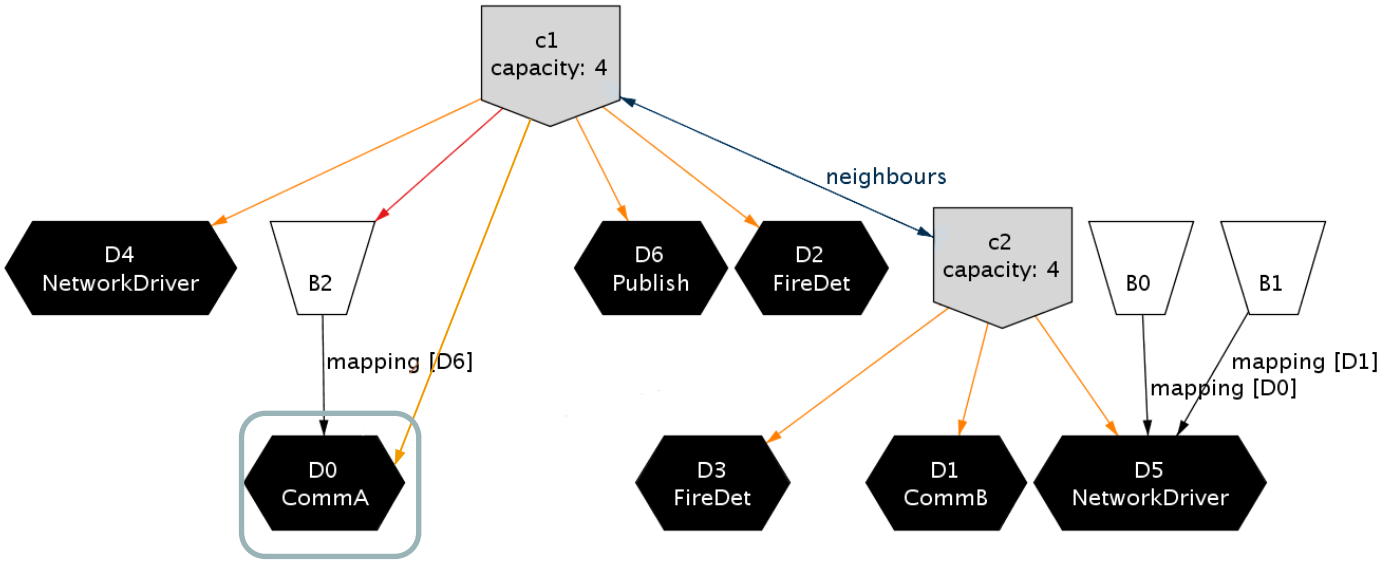
NetworkDriver's StateMachine (**SM1**) initializes **D5**'s reconfiguration from **D0** to **D1**

Alloy Trace Sequence No4



Alloy selects **Migrate** and starts **D0**'s migration from **c2** to **c1**

Alloy Trace Sequence No5



CommA (D0) has finished migrating to **c1**, c1's capacity is still 4