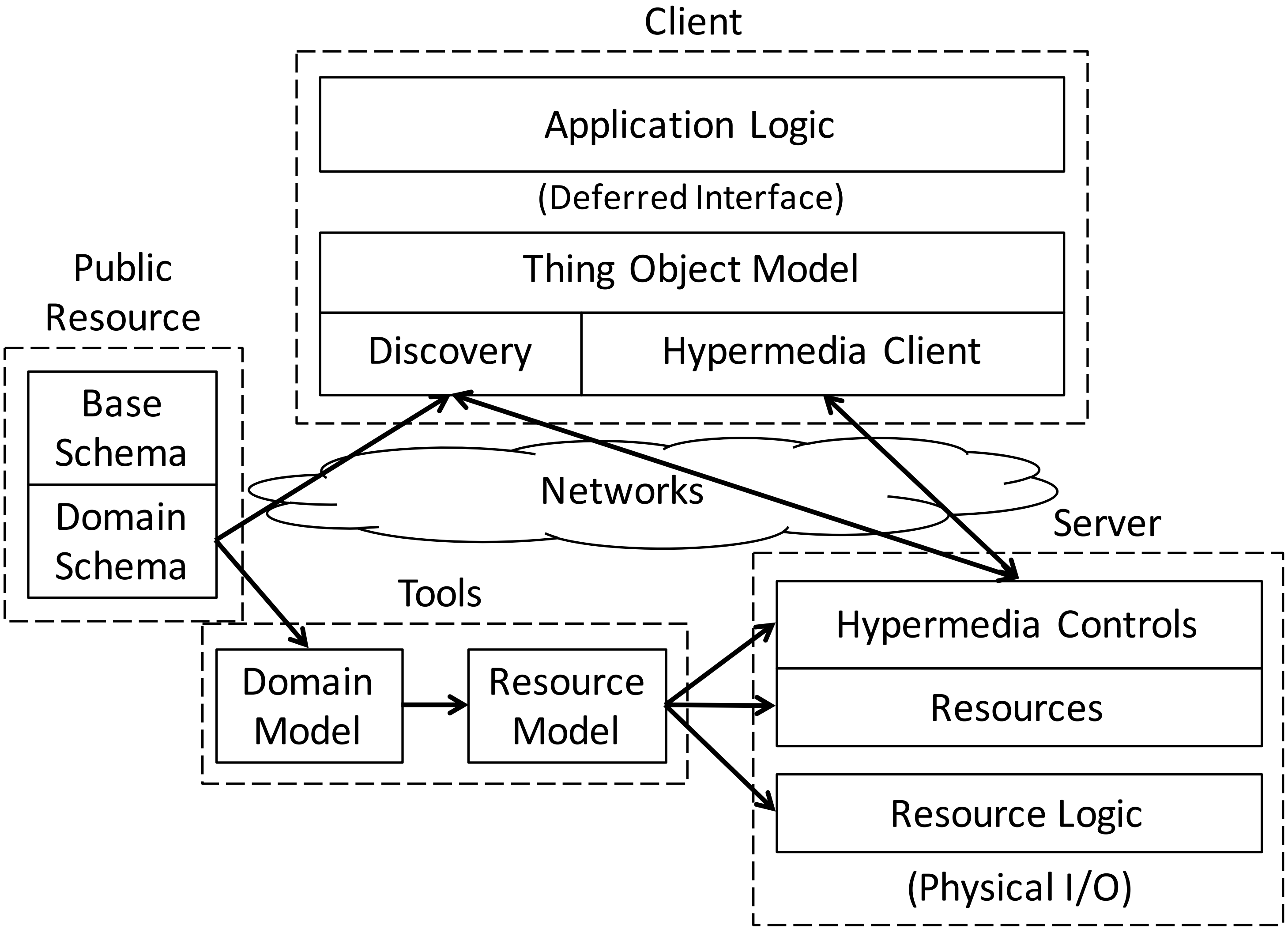
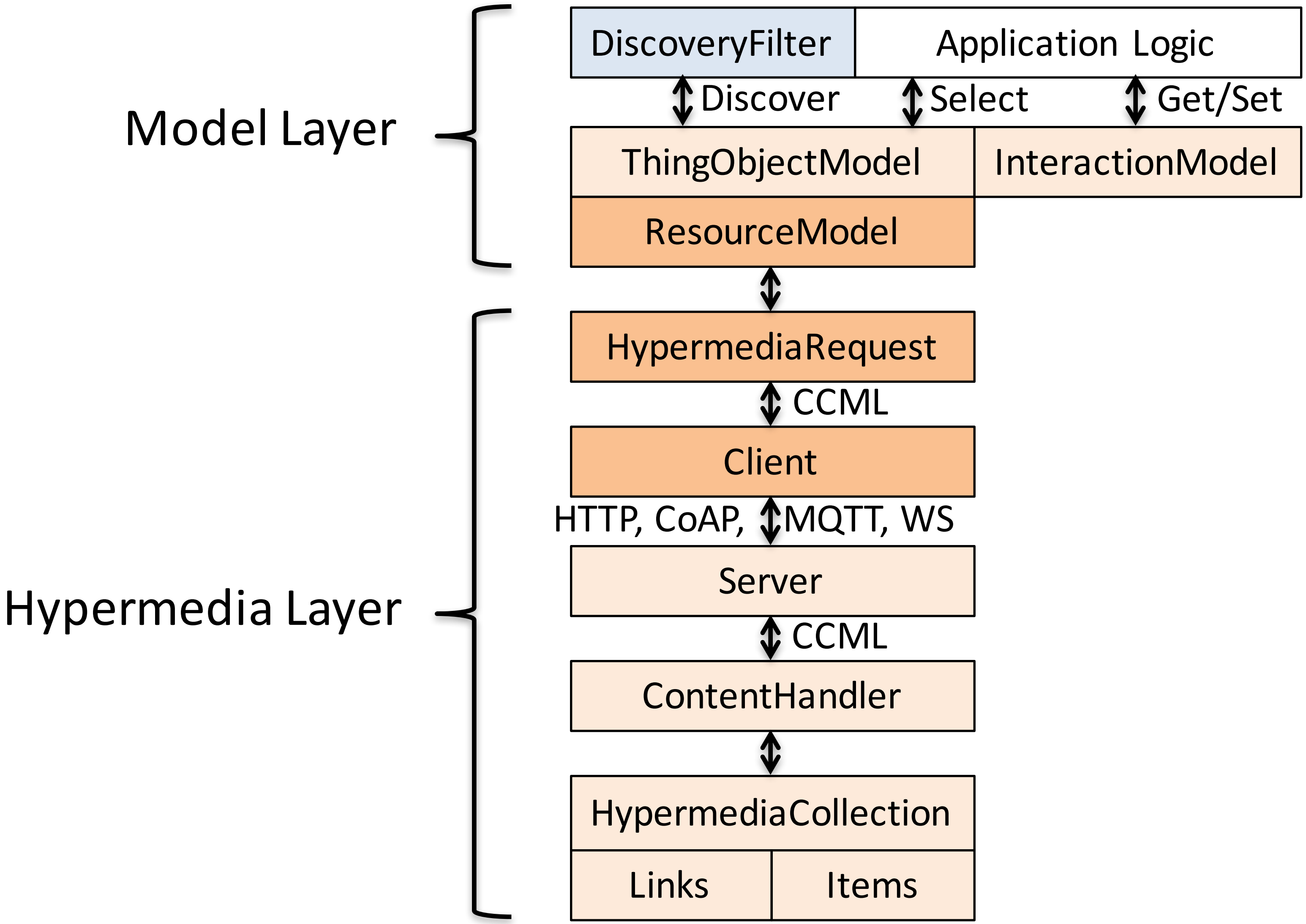
Machine Hypermedia Toolkit

Reference Implementation and System Demonstrator

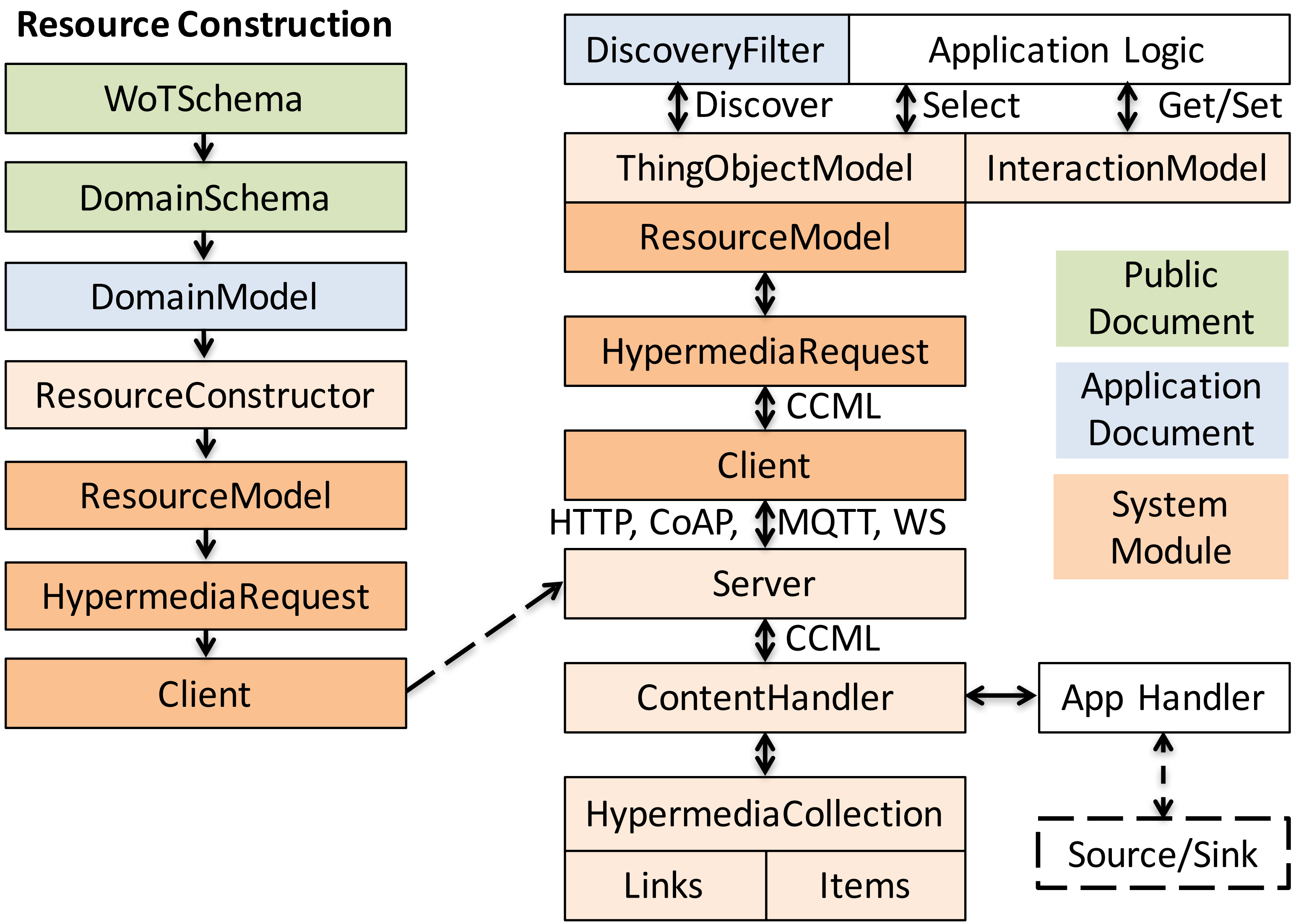
Hypermedia System Architecture



Demonstrator Architecture



Demonstrator Architecture



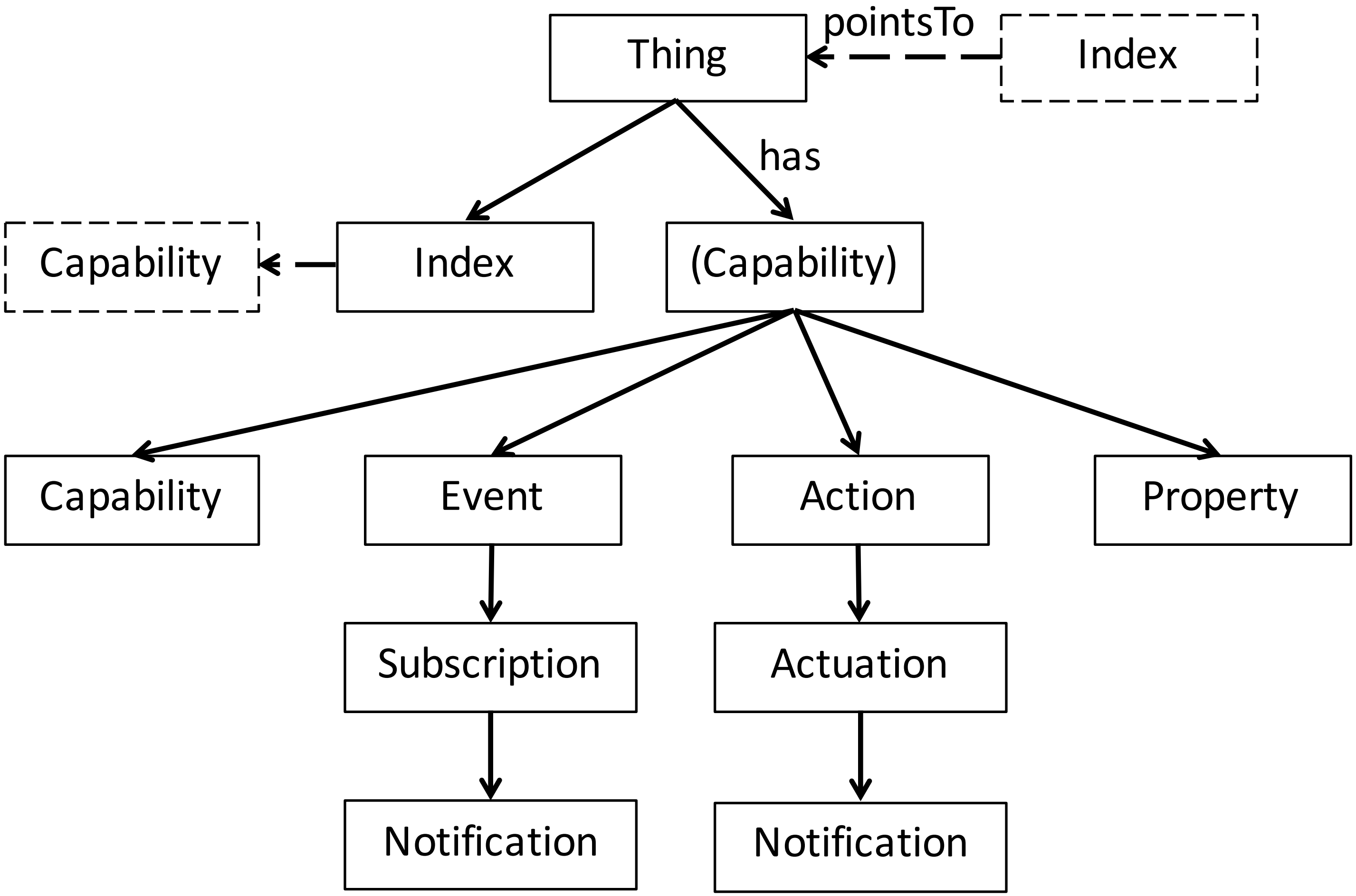
Thing Object Model

- Thing Object Model (TOM) is a container for a set of resource references
- Application may use the domain schema to understand terms describing TOM resources
- Provides discovery and interaction model abstractions in the form of a scripting API
- Client State Machines, Discovery, and Form Logic

Interaction Model

- Interaction model is a set of functional abstractions that enable application state machines to be standardized
- The W3C Interaction model consists of Properties, Events, and Actions
- Properties are resource elements
- Actions and Events are behaviors
- The TOM maps actions and events to hypermedia forms and POST operations

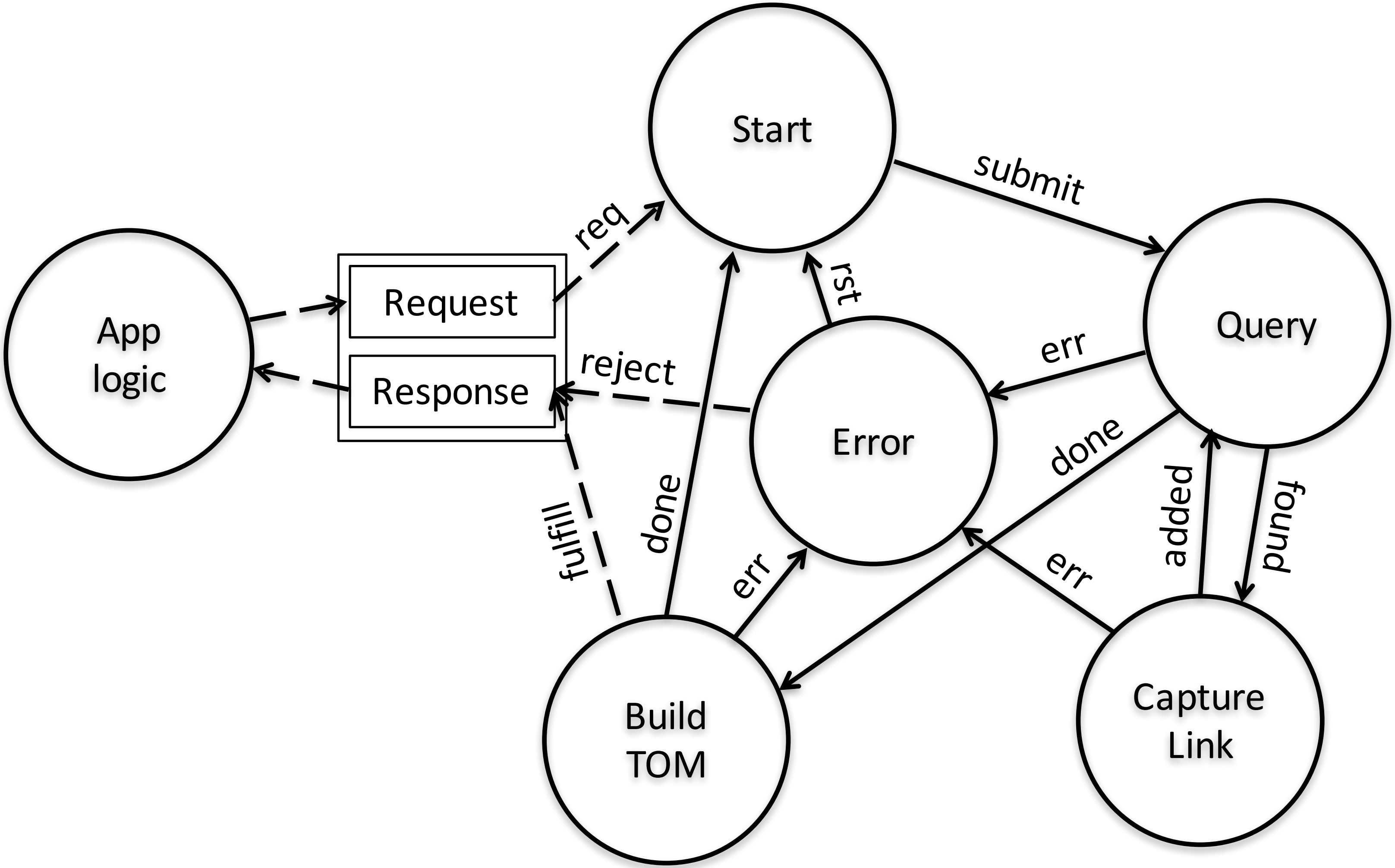
Interaction Model Relationships



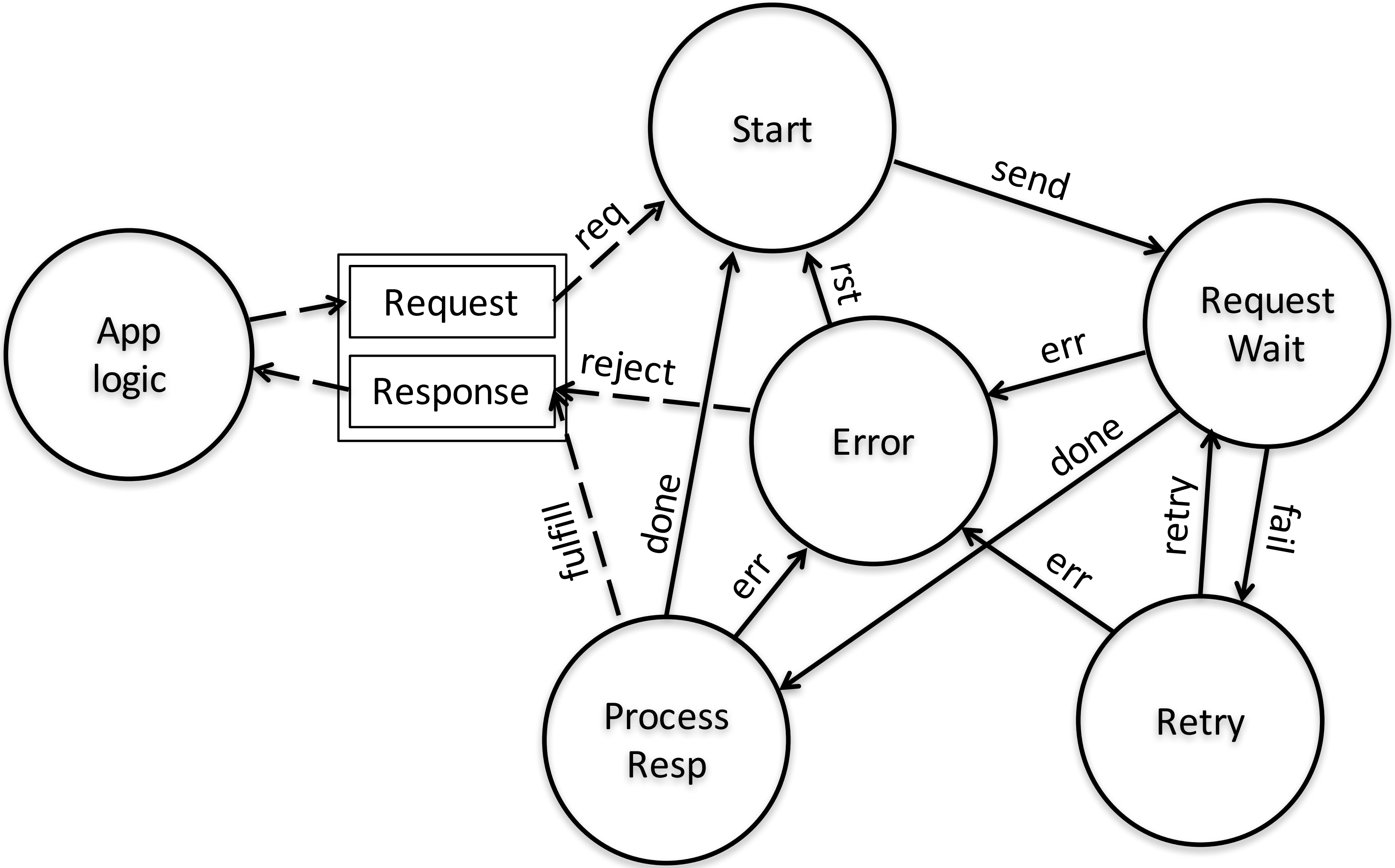
Example Model

```
class: brightness,
type: capability,
description: "brightness control"
usedBy: [ light ],
mayHave: [
   currentBrightness, targetBrightness,
   stepBrightness, moveBrightness,
   change, step, move, stop,
   propertyValueChange ],
params: {_
   targetValue: _targetBrightness,
   _stepSize: _stepBrightness,
   _moveRate: _moveBrightness},}
```

State Machine - Discovery



State Machine - Interaction



TOM Discovery and Interaction

```
 tom = ThingObjectModel()
    print "discovering"
    tom.addGraph("http://10.0.0.45:8000/index/" ).discover( _filter)
    print "completed"

    """ current-level  is type Property so will have get and set methods
bound to it. server request is sent and model value is updated. Value is
returned on get. Get the current value, update with an incremented value,
and read back """

    startingLevel = tom.byLabel("current-level").get()
    print "starting level:", startingLevel
    newLevel = startingLevel + 10
    if newLevel > 100:
        newLevel = 0
    tom.byLabel("current-level").set(newLevel)
    print "new level:", tom.byLabel("current-level").get()
```

Discovery Template

```
_filter = [
  {
    v._rt: d._light,
    v._label: "mylight",
    v._has: [
       {
         v._rt: d._brightness,
         v._label: "dimmer-control",
         v._has:[
            {
              v._rt: d._currentbrightness,
              v._label: "current-level"
            }
         ]
       }
    ]
  }
]
```