



A Tool Builders Perspective

Max Ott



Australian Government
**Department of Broadband, Communications
and the Digital Economy**
Australian Research Council

NICTA Funding and Supporting Members and Partners



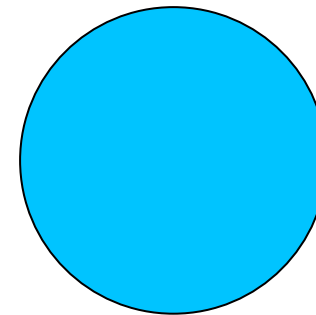
Users, Resources, Tools



Users care about resources (and only about resources)



User

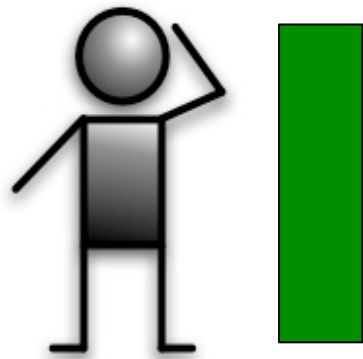


Resource

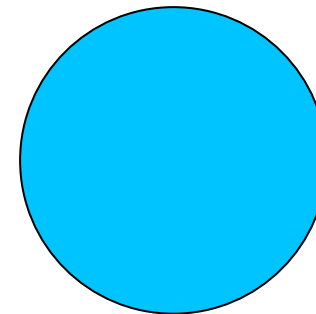
Users, Resources, Tools



Tools help users to provision resources and orchestrate experiments



Request →

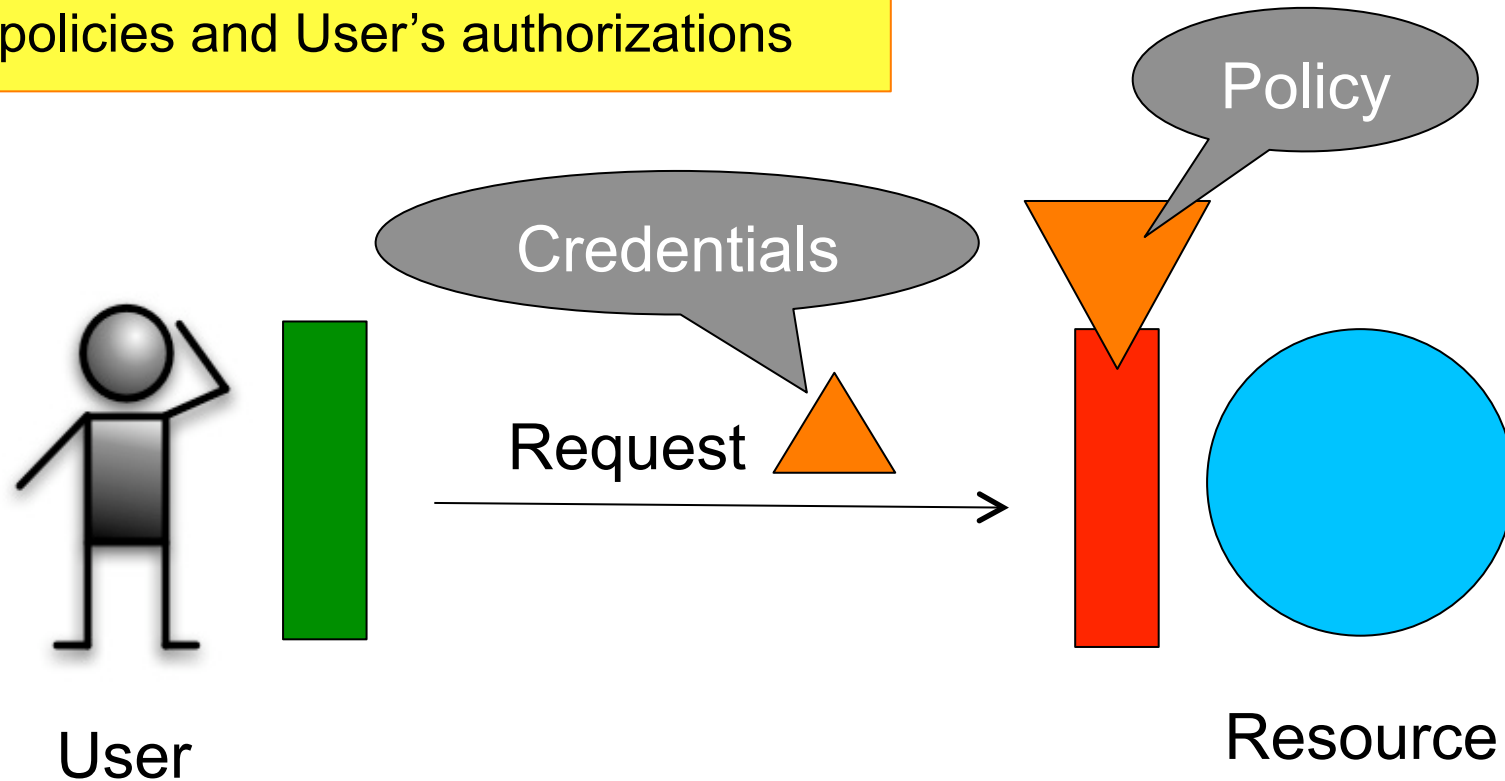


User

Resource

Users, Resources, Tools

User access is governed by Owner's policies and User's authorizations



There are many types of Resources



- VM
- VLAN, VPN
- Frequency, Channel
- Sensors, cameras, motes
- Spectrum analyzer
- Robots
- Applications, services
- Measurement collection

What makes my life easier



- A realistic expectation
 - Everything can and will break, but not too often
 - Everything is best effort, but everyone is trying hard
- A few concepts applied consistently
 - A simple resource model (entities, properties, values)
 - A resource grouping mechanism
 - A resource life cycle (create - modify - release)
 - An accountability mechanism

A simple resource model



Resources have named properties which have values or contain references to other resources.

- Resource model is generic
 - We should agree on how to ‘encode’ that
- Semantic meaning of properties and values is domain specific
 - We may even agree on how to formally define them

A resource grouping mechanism



- When doing experiments at scale we do NOT want to deal with individual resources
- A 'group' is a resource which contains resources
 - Can also include other groups (hierarchical)
- Operations on a group are recursively applied to every member in that group (and sub-groups)
 - Exact semantics needs to be defined

Groups Applied

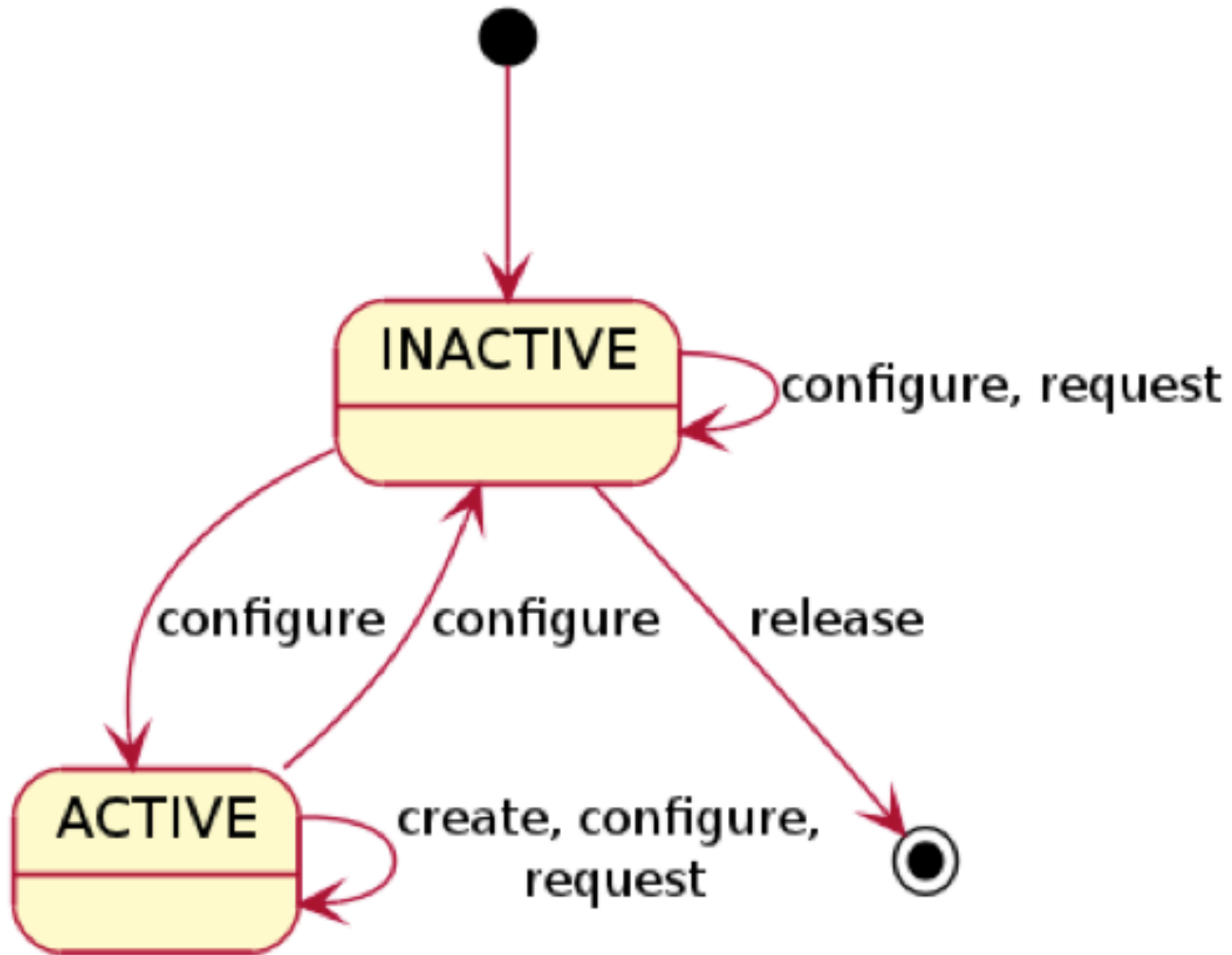


```
defTopology 'mf-demo1' do |t|
  allResources.select(... Some query ...).each_with_index do |n, i|
    props = {:guid => i, :self_ip => "192.168.1.#{i}"}
    t.add n, :groups => [:routers], :props => props
  end
  ...
end
```

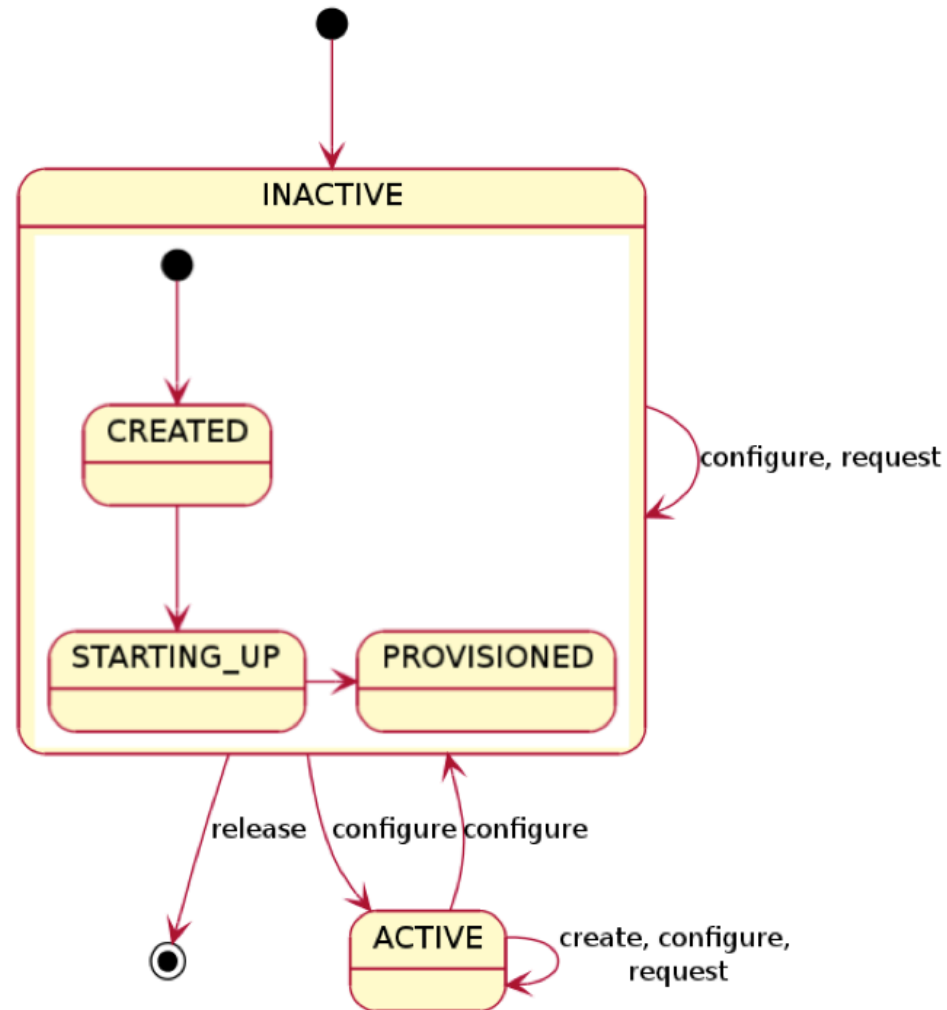
```
defGroup "senders" do |g|
  g.addApplication 'MF-Linux-File-Sender-App' do |app|
    app.setProperty 'file', file_to_send
  ..
  end
end
```

```
defGroup 'receivers' do |g|
  ...
end
```

A resource life cycle



... resource-specific extensions



Translates to a SIMPLE Protocol



- Create
 - put
- Configure
 - post
- Inquire
 - get
- Release
 - delete
- (Inform)

An accountability mechanism

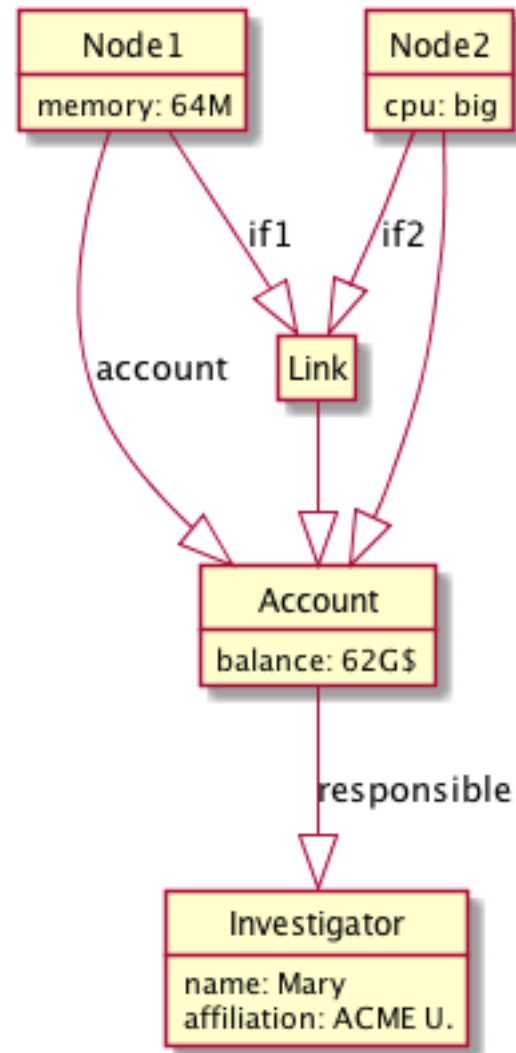


- Each resource belongs to exactly ONE “account”
 - Account is synonym for slice or project
- Accounts are the basis of resource allocation
- Accounts have exactly ONE accountable Person
- Accounts can belong to other accounts

Tying things together

entity property value

- Node1 memory 64M
- Node1 connected_to Link
- Node1 account Account
- Account owned_by Investigator
-

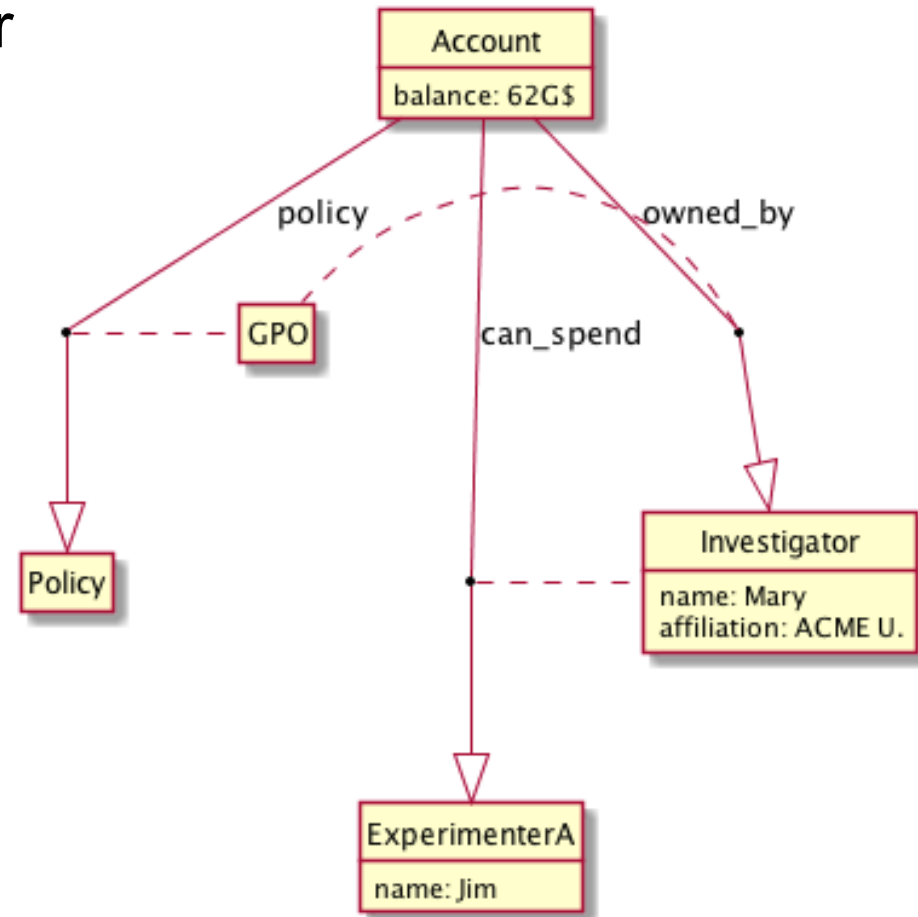


Tying things together



entity property value assenter

Account balance 62G\$ says Bank
Account owned_by PI says GPO
Account policy Policy says GPO
ExpA can_spend Account says PI



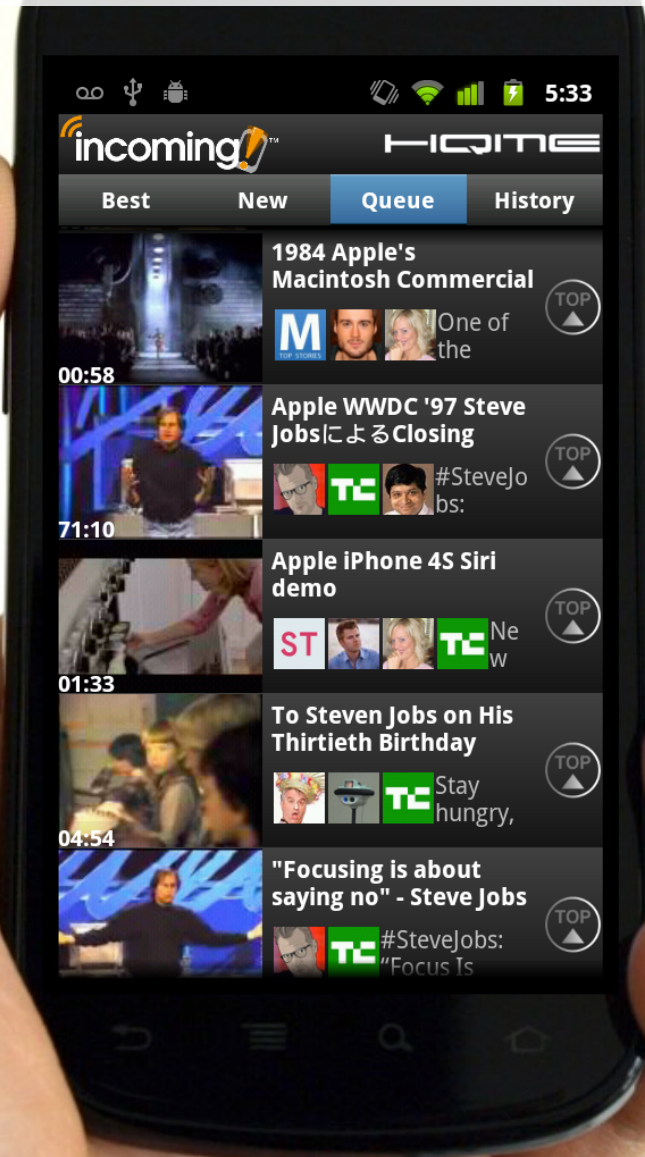
What makes my life easier



- A realistic expectation
 - Everything can and will break, but not too often
 - Everything is best effort, but everyone is trying hard
- A few concepts applied consistently
 - A simple resource model (entities, properties, values)
 - A resource grouping mechanism
 - A resource life cycle (create - modify - release)
 - An accountability mechanism

<http://omf.mytestbed.net/projects/omf/wiki/ArchitecturalFoundation2>

The Action is at the Mobile Edge



Observing user behavior

Observing local environment

Observing social network activities

