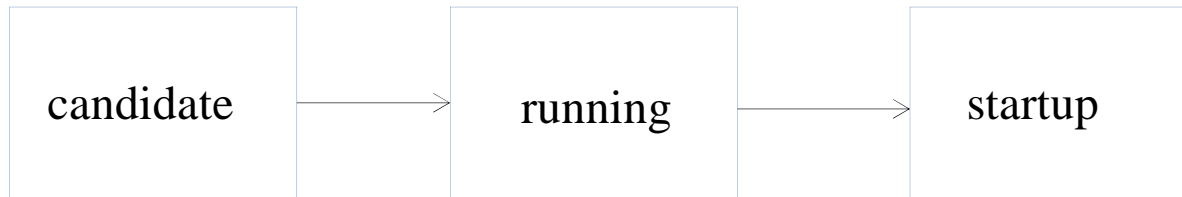


I2RS Protocol Simple Example

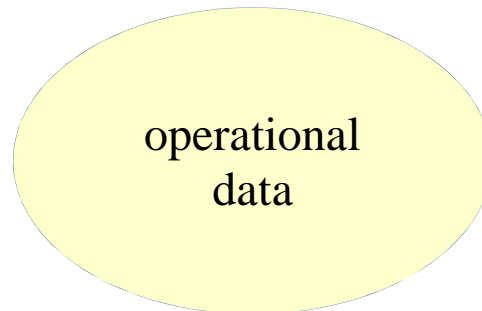
Hares work expansion of
Andy Bierman, Kent Watsen work

Previous Definitions



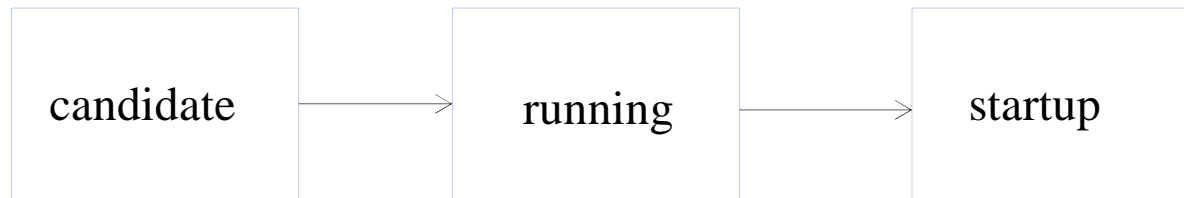
config true;

config false;



All operational data exists alongside config=true but there is no datastore defined for config=false data nodes

Definitions from ietf-netmod-opstate-req



config true;

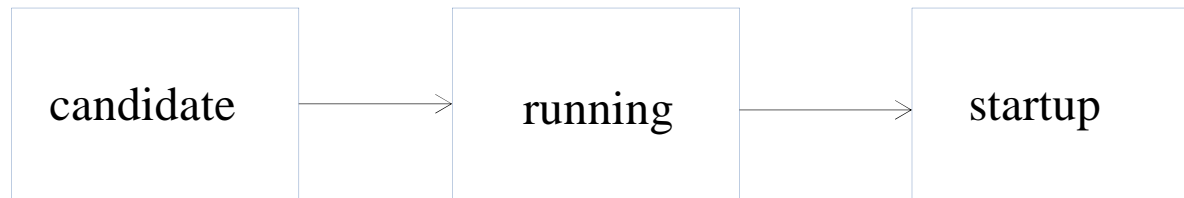
intended config

config false;

applied config

Derived state

Definitions from ietf-netmod-opstate-req



config true;

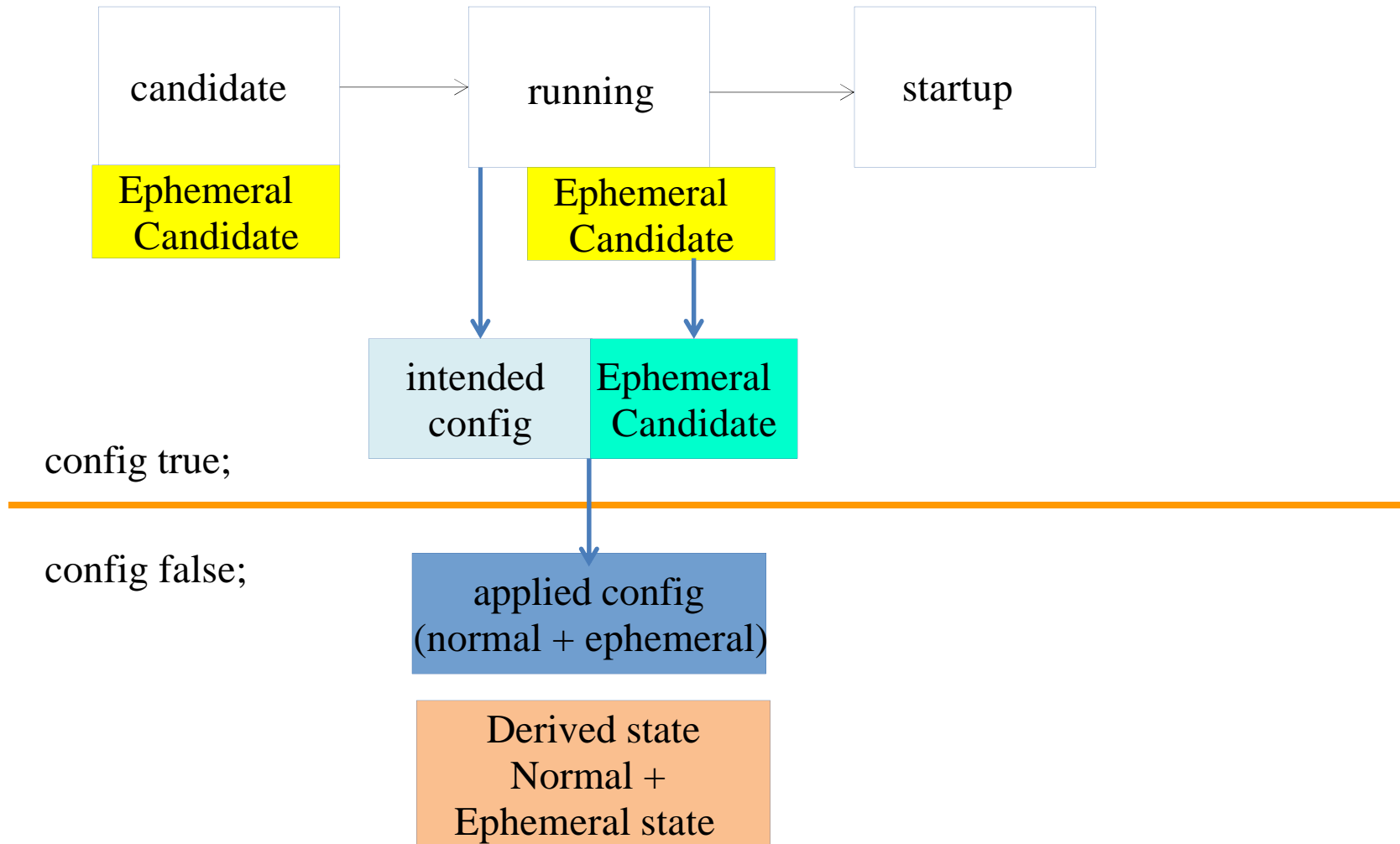
intended config

config false;

applied config

Derived state

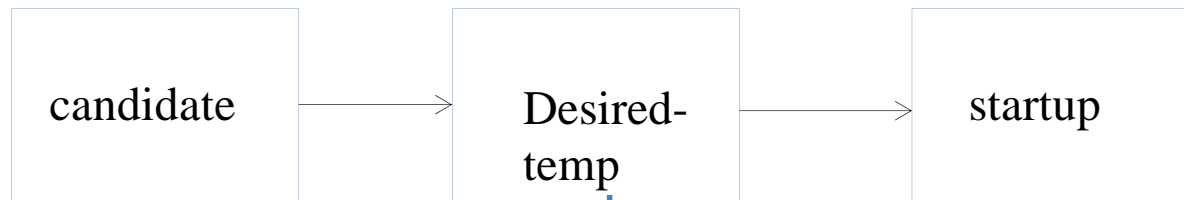
Ephemeral Additions



Simple Thermostat Example

```
module thermostat {  
    ...  
    leaf desired-temp {  
        type int32;  
        units "degrees Celsius";  
        description "The desired temperature";  
    }  
  
    // operational state  
  
    leaf actual-temp {  
        type int32;  
        config false;  
        units "degrees Celsius";  
        description "The measured temperature";  
    }  
}
```

Thermostate Model



config true;

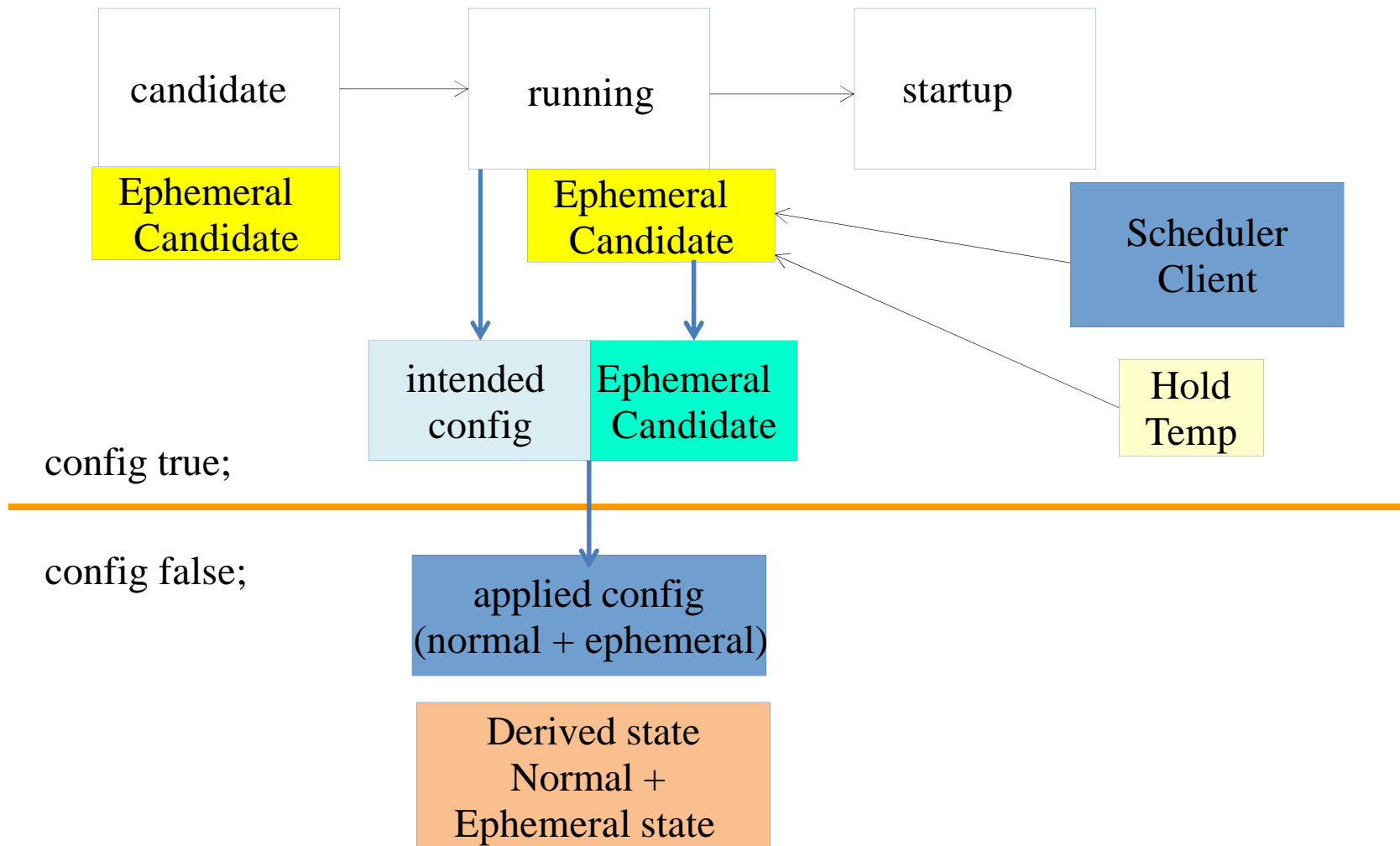
Intended Config
Desired-temp

config false;

applied config
Desired-temp

Derived state
Actual-temp

Thermostat + I2RS



Simple Thermostat + ephemeral

```
module thermostat {  
  ...  
  leaf desired-temp {  
    type int32;  
    ephemeral true;  
    ephemeral-validation full-check;  
    units "degrees Celsius";  
    description "The desired temperature";  
  }  
}
```

Operational State:

```
  leaf actual-temp {  
    type int32;  
    config false;  
    units "degrees Celsius";  
    description "The measured temperature";  
  }  
}
```

RESTCONF Example

RESTCONF Running Datastore Edit

```
PUT /restconf/data/thermostat:desired-temp
```

```
{ "desired-temp": 18 }
```

RESTCONF Ephemeral Datastore Edit of config=true

```
PUT /restconf/data/thermostat:desired-temp?context=ephemeral
```

```
{ "desired-temp": 18 }
```

NETCONF

```
<rpc-message-id=101>
  <xmlns="urn:ietf:params:xml:ns:base:1.0">
    <edit-config>
      <target>
        <ephemeral>
          True
          <ephemeral-validation>
            full-check
          </ephemeral-validation>
        </ephemeral>
      </target>
    <config>
      <top xmlns="http://example.com/schema/1.0/thermostat/config">
        <desired-temp> 18 </desired-temp>
      </top>
    </config>
  </xmlns>
</edit-config>
</rpc-message-id=101>
```

Capability Specification for NETCONF/RESTCONF

NETCONF (1)

- **Capability: ephemeral-datastore**
- *Overview:*
 - Not intended to survive a reboot, and Never locked
 - 1 Pane of glass ephemeral vs config 1 pane of class: Last Write wins (alterable via operator-applied policy)
 - Multiple clients writes to 1 Pane of ephemeral glass = error, but highest priority wins for ephemeral glass
 - Signaled as capability for node, grouping, sub-model, model via NETCONF <hello> - but no non-ephemeral under ephemeral modules, sub-trees, node
 - Ephemeral error checking: 1)syntax only, 2) reduced, 3) full-check
 - Yang statement “ephemeral”
- **Dependencies:**
 - Yang: ephemeral flag, ephemeral-validation
 - Yang modules – must support notification of write conflicts (Config/ephemeral and Priority)

NETCONF (2)

- New operations :
 - Link-ephemeral <target-config>
 - Bulk-write – [Not sure if need or if rpc better approach]
- Modifications
 - <get-config> <get> - target changes
 - <edit-config> - <merge-priority> <replace-priority>
 - <default-operations>: <merge-priority> or <replace-priority>
 - <error-option> - “all-or-nothing” == “rollback-on-error”
 - <unlock> <lock> - not supported
 - <confirmed commit> - not supported
 - <close-session> <kill-session> - target change
 - <Writable-running> and <candidate> – support ephemeral (?)
 - Validate – supports ephemeral data store with three key words:
Syntax, reduced, full-check

RESTCONF (1)

- **Capability: ephemeral-context**
- *Overview:*
 - Not intended to survive a reboot, and Never locked
 - 1 Pane of glass ephemeral vs config 1 pane of class: Last Write wins (alterable via operator-applied policy)
 - Multiple clients writes to 1 Pane of ephemeral glass = error, but highest priority wins for ephemeral glass
 - Signaled as capability for node, grouping, sub-model, model via NETCONF <hello> - but no non-ephemeral under ephemeral modules, sub-trees, node
 - Ephemeral error checking: 1)syntax only, 2) reduced, 3) full-check
 - Yang statement “ephemeral”
- **Dependencies:**
 - Yang: ephemeral flag, ephemeral-validation
 - Yang modules – must support notification of write conflicts (Config/ephemeral and Priority)
 - I2RS Yang modules support: Yang patch and Yang module library

RESTCONF (20)

- Data resources
 - +restconf/data – ephemeral data tree with edit collision features of timestamp and Entity Tag
 - Assumption: Entity can be split to client-priority
- Modifications
 - Options: provide indication of ephemeral state in data modules, sub-modules [ietf-netconf-yang-library]
 - HEAD – returns ephemeral or config context
 - GET - determines if ephemeral or config
 - POST/PUT/PATCH - context=ephemeral:
uses ephemeral rules + validity + priority + no config below ephemeral
 - DELETE – ephemeral context
 - Query – Allows to filter by ephemeral
 - Error/Notifications – must interact with pub/sub push [ietf-netconf-yang-push]
 - Log and traceability -

RESTCONF

- capability-name: ephemeral-datastore
- Dependencies: