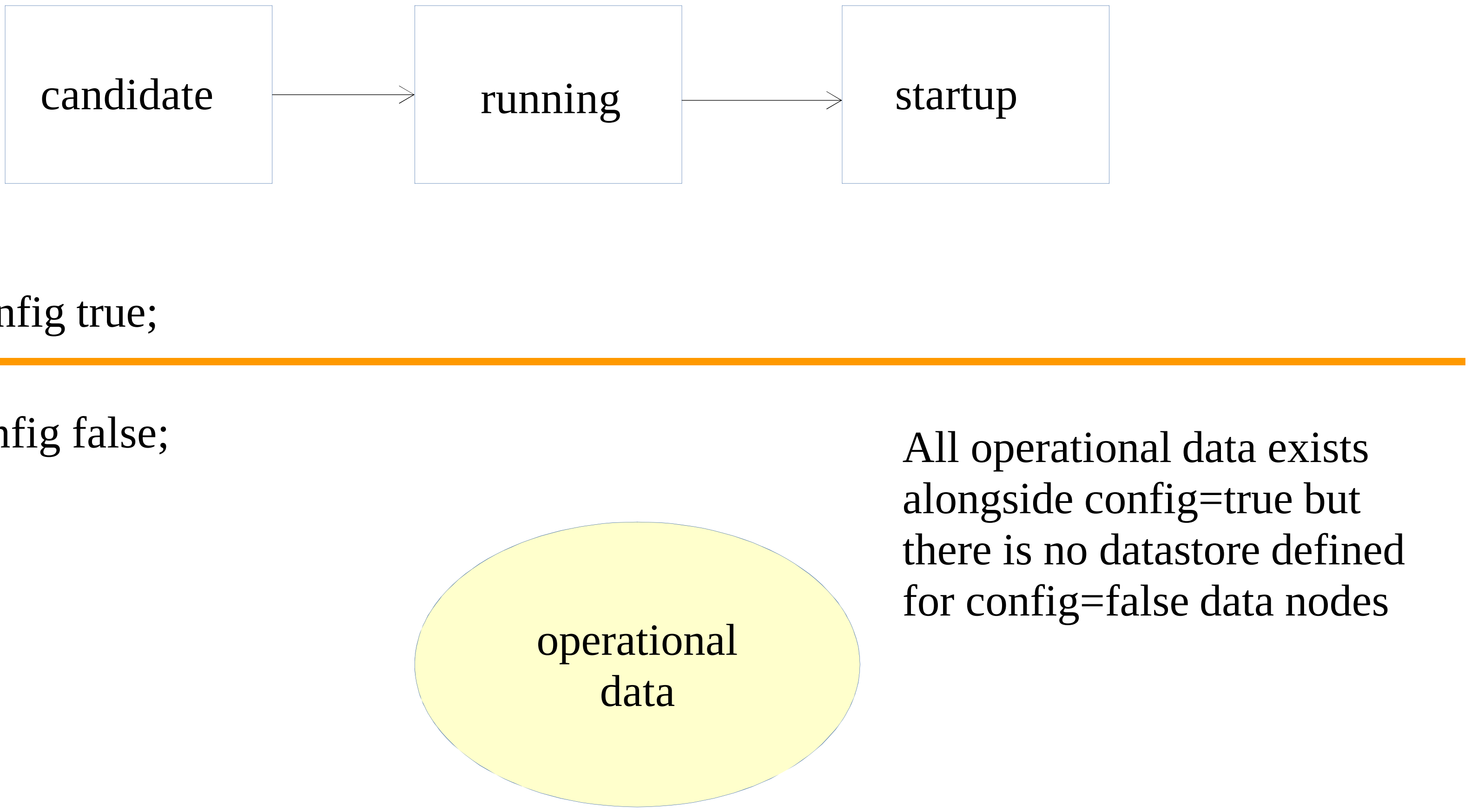
I2RS Protocol DT Meeting

9/4/2015

Extended Datastores (3)

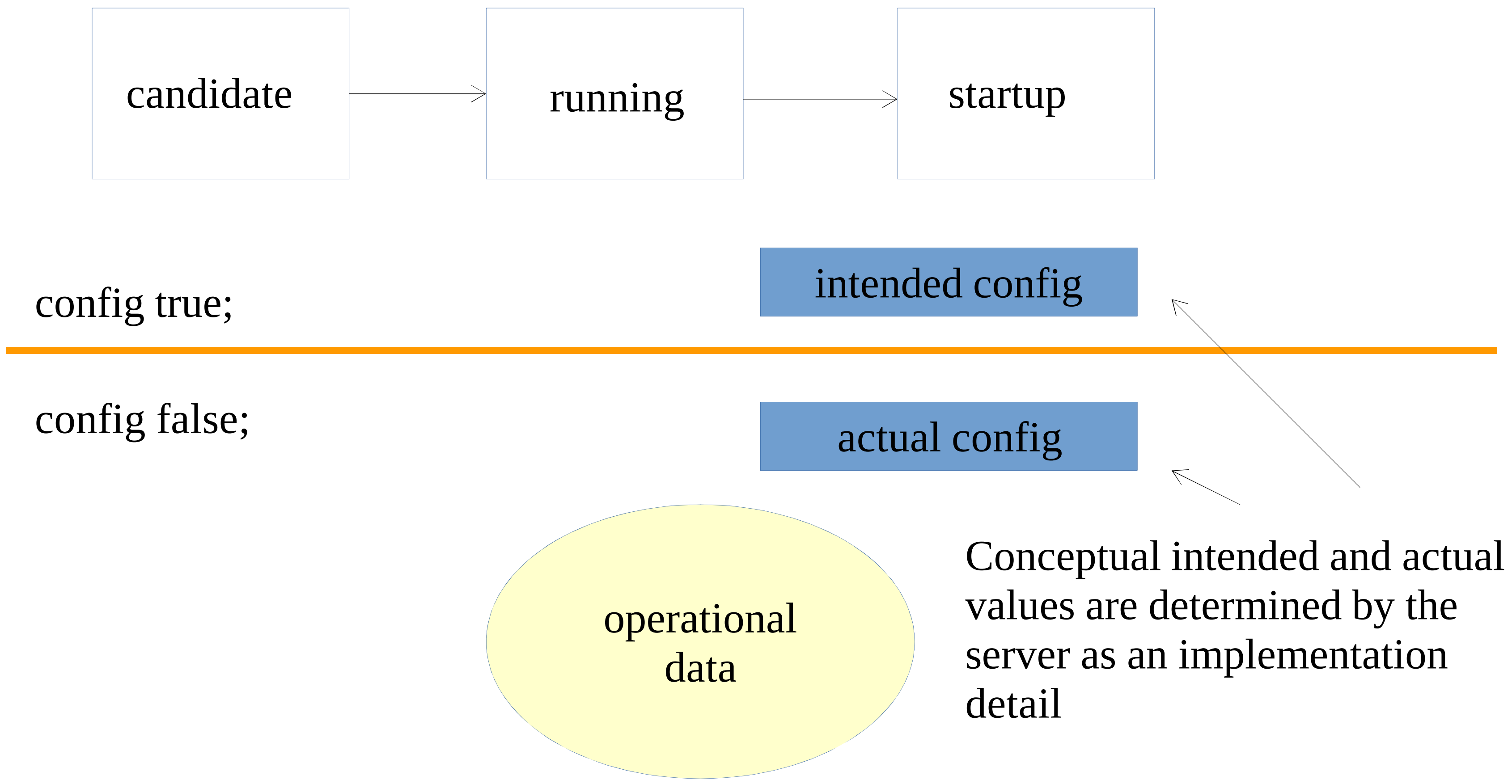
- 3rd attempt to converge Kent's slide with the thermostat example
  - Andy Bierman <andy@yumaworks.com>
    - 31-AUG-2015

Current Datastores



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

Current Datastores (Ext. 1)



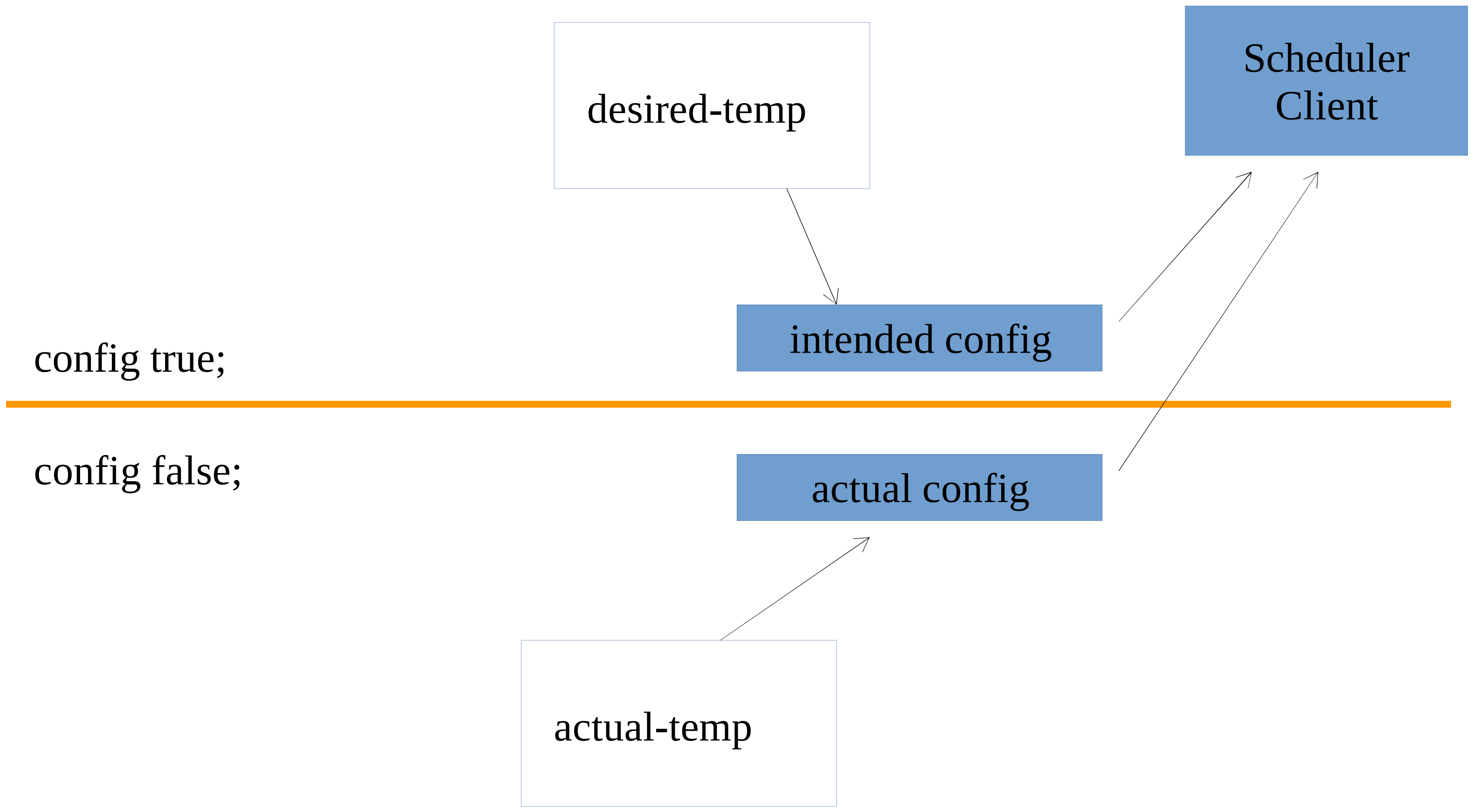
Conceptual intended and actual values are determined by the server as an implementation detail

Simple Thermostat Example

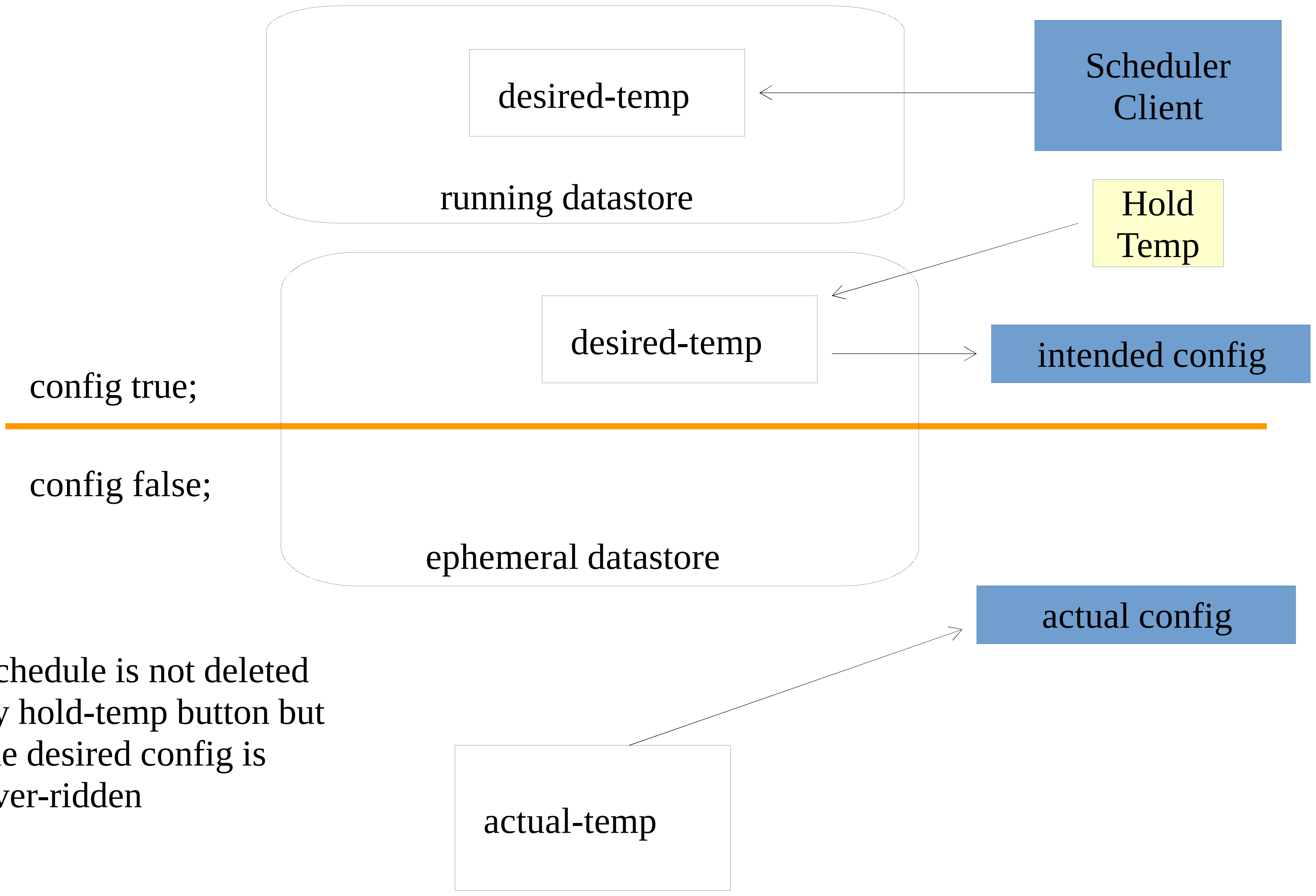
```
module thermostat {
    …
    leaf desired-temp {
        type int32;
        units “degrees Celsius”;
        description “The desired temperature”;
    }

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

Thermostat Model

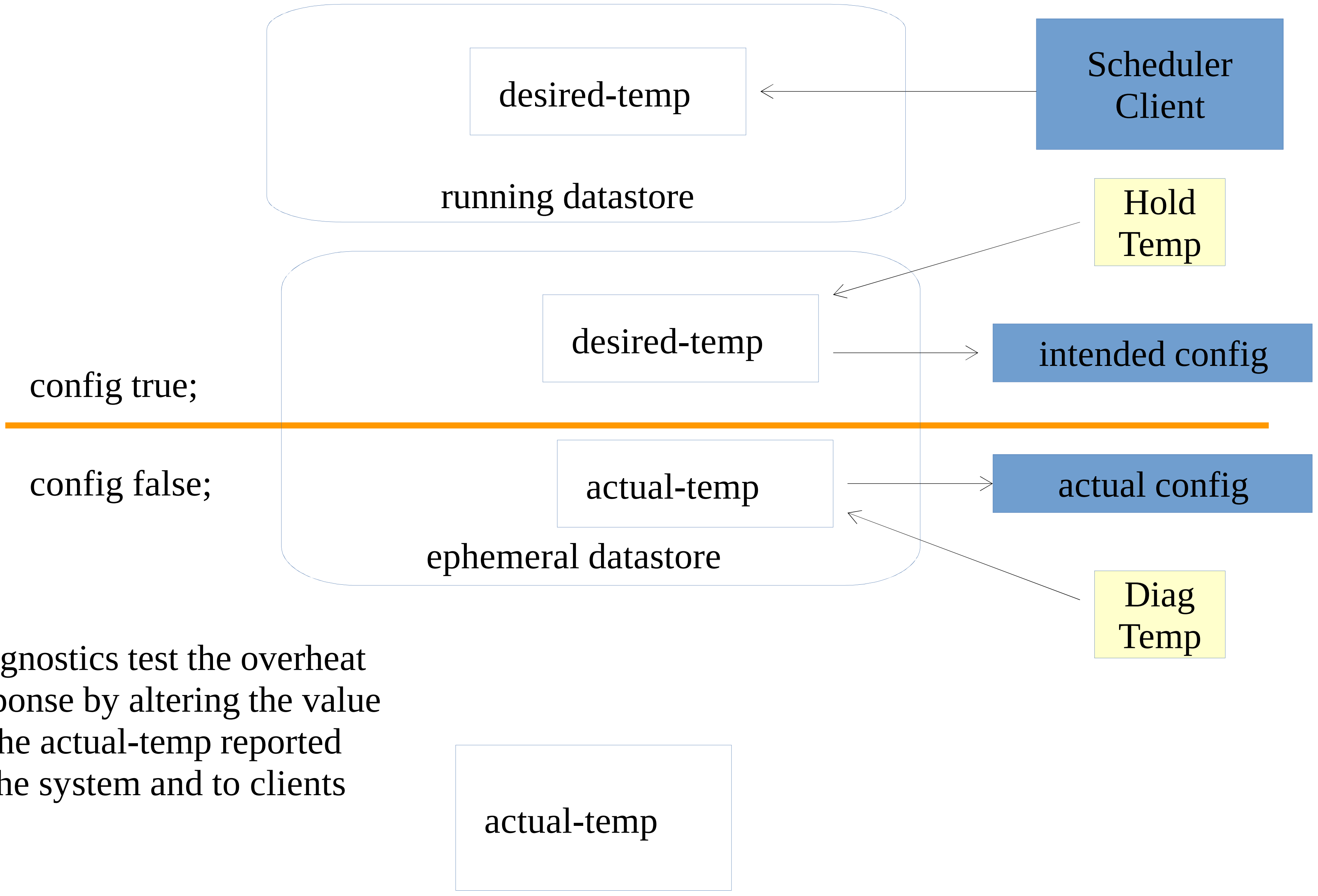


Thermostat Model + Hold Temp



Schedule is not deleted by hold-temp button but the desired config is over-ridden

Thermostat Model + Diagnostics



Diagnostics test the overheat response by altering the value of the actual-temp reported to the system and to clients

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?datastore=ephemeral

{ “desired-temp”: 18 }

RESTCONF Ephemeral Datastore Edit of config=false

PUT /restconf/data/thermostat:actual-temp?datastore=ephemeral

{ “actual-temp”: 72 }

Issues

- config=true easy to implement
  - System can use the same instrumentation edit API to send ephemeral value instead of running config value
    - OK to say that any config=true node can be edited in the ephemeral datastore
    - I2RS must be identified in the yang module
  - config=false hard to implement
    - The operational state does not have an “edit API”
    - Server can only be expected to add this API for specific objects where use-cases exist
      - how to identify config=false nodes that are allowed to be edited in the ephemeral datastore?

Simple Thermostat + ephemeral

```
module thermostat {
    …
    leaf desired-temp {
        type int32;
        units “degrees Celsius”;
        description “The desired temperature”;
    }

    leaf actual-temp {
        type int32;
        config false;
        ephemeral true;
        units “degrees Celsius”;
        description “The measured temperature”;
    }
}
```

Need to identify this leaf as OK to edit in the ephemeral datastore

Summary

- Both config=true and config=false nodes can be edited in the ephemeral datastore
  - this datastore overrides normal intended config and actual config (implementation details)
- Edit and validation rules for ephemeral datastore can be different than for the running datastore
  - Actual rules TBD but cannot reference data that is “less stable” than the current context
  - Want to minimize performance overhead; maybe even provide mode where YANG validation rules are skipped