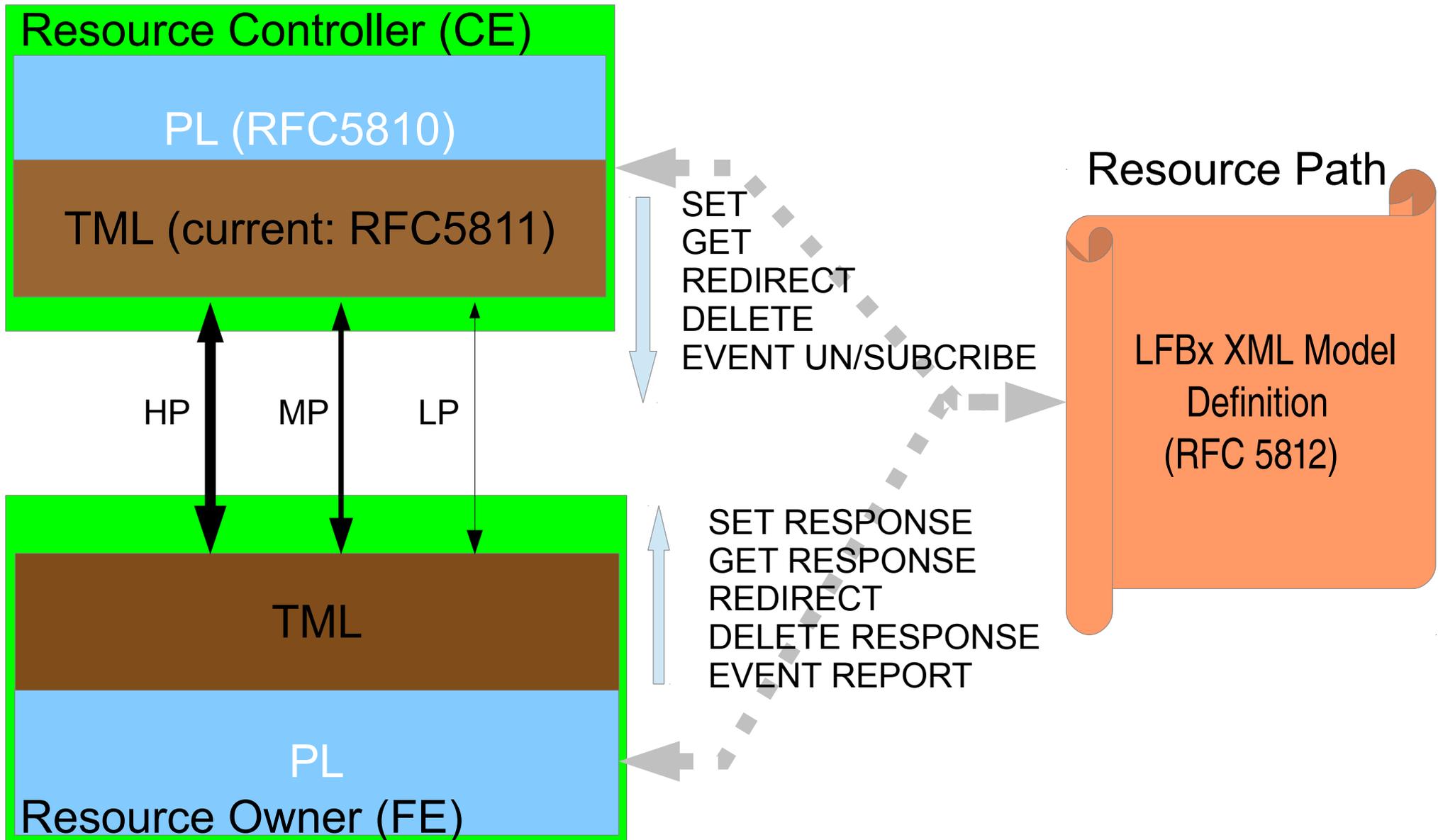


ForCES Model Gap Analysis for I2RS

Jamal Hadi Salim <hadi@mojatatu.com>
IETF 89, London, UK

ForCES Architecture In A Nutshell



ForCES Architecture In A Nutshell

- A protocol (The *Verbs*)
 - A modular transport for the protocol
- A data model (The *nouns describing resources*)
 - Logical Functional Block which are constructs that describe the resource
- Combine the above and you have a language
 - [*<verb> <noun> [args]*]⁺
 - Anti-RPC
 - Few verbs but infinite possibilities of nouns

LFB Class Definition

Datatype definition

LFB Definition

Components Describing Resources
(use *Datatype* Definitions)

Resource Capabilities
(Using *Datatype* Definitions)

Events monitoring and reporting on *Components*

LFB Class

- Object oriented resource definition
- Each class has definitions for:
 - Datatype, components, Capabilities, Events
- Multiple instances of an LFB class can be created/instantiated.
 - Example: Class Rib instance 2
 - Each class instance has its own:
 - State/config
 - capabilities
 - events

LFB Datatype Definitions

- Formal constraints for validation of defined attributes
- Atomic types, complex/compound types,
- grouping of compound types in the form of structures and indexed/keyed tables
- Hierarchical/tree semantics
- Aliasing to symlink shared infrastructure
- Optionality and default values
- Basic ACL (RW permissions)

LFB Class Definitions

- Components
 - data type definitions of control/config/state resource attributes acted on by a controller via the ForCES protocol
- Capability
 - definitions of resource capabilities and capacities advertised by the resource owner
- Events
 - hooks for publish/subscribe with expressive trigger and report definitions
 - count, threshold which could be binary, range, or time which could be formed into a compound expression using and/or operators

LFB Class Extensibility

- Inheritance and extension of a parent class
- Inheritance and extension of data definitions
- Backward and forward compatibility of LFB classes and defined data structures
 - Versioning
 - Be liberal in what you expect and conservative in what you do

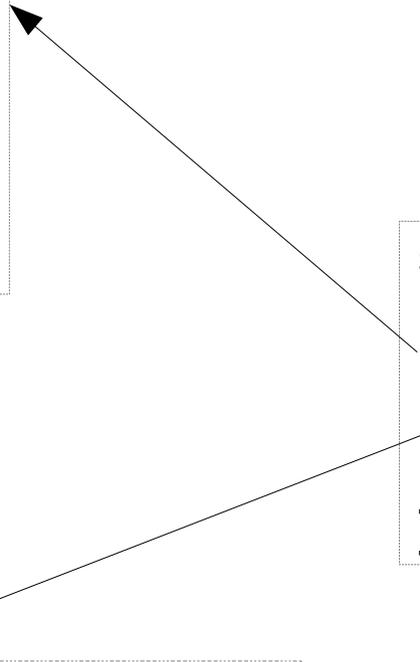
Example DataType

```
enum rib-type: {  
  IPV4_RIB_FAMILY  
  IPV6_RIB_FAMILY  
  MPLS_RIB_FAMILY  
  IEEE_MAC_RIB_FAMILY  
}
```

```
struct rib: {  
  RIB_NAME string[16],  
  rib-family rib-type,  
  Routetable array of type route,  
  boolean ENABLE_IP_RPF_CHECK  
}
```

```
struct route: {  
  Match of type matchtype,  
  nexthops array of struct nexthop-list,  
  Optional table of route-attributes  
  Optional table of route-vendor-attributes  
}
```

```
union match : {  
  ipv4-route  
  ipv6-route  
  mpls-route  
  Mac-route  
  interface-route  
}
```



Example components

- component id 1:
 - INSTANCE_NAME type ***string[N]***, *read-write*
- component id 2:
 - ROUTER_ID type ***uint32***, *read-write*
- component id 3:
 - *optional* interface-list array of type ***ifindex***, *read-write*
- component id 4:
 - rib-list array of type ***rib***, *read-write*

Example Capabilities

- capability id 27:
 - NH_CHAIN_DEPTH type uint16

Example Events

- Event id 1: monitor Routes table,
 - *advertise route that changed*
- Event id 2: monitor Routes table,
 - *advertise route added*
- Event id 3: monitor Routes,
 - *advertise route deleted*
- Event id 4: monitor Nexthop resolution,
 - *advertise nexthop + changed status*
 - (state: resolved/unresolved)

Gaps

- Overhead in table dumps or bulk sets when tables have “holes”
 - Requires use of ILV per table row (64 bit overhead)
 - Could be burdensome if you have small table rows (less than 64 bits in total size)
 - Does not seem to be an issue relative to the RIB information model

Gaps

- New Data type definitions maybe needed for RIB info model
 - List datatype
 - Worrisome is when list elements are not the same size
 - “At least one of these”
 - Current approach is to tag all but one element as non-optional
- Union base types may require some rethinking
 - Needed by some of the RIB information model
 - Refer to discussion: <http://www.ietf.org/mail-archive/web/forces/current/msg04668.html>

Pros/Cons

- Pros

- Extremely extensible and simple programmatic Interfaces

- Cons

- Small Changes required to fully meet I2RS spec