ForCES Protocol Gap Analysis for I2RS

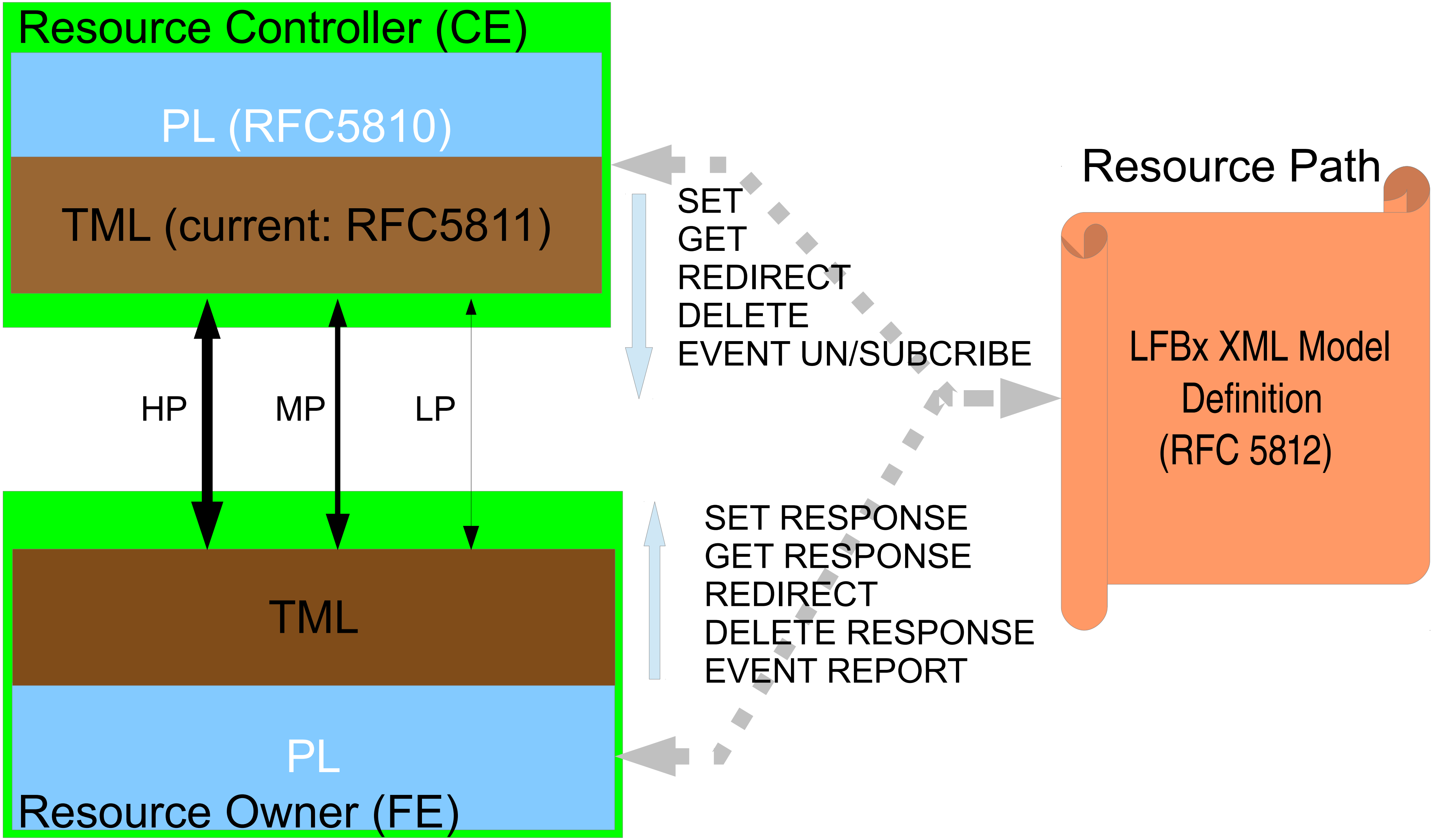
Jamal Hadi Salim <hadi@mojatatu.com>

IETF 89, London, UK

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

ForCES Architecture In A Nutshell



Protocol Semantics

- Transport independence
- Simple Verbs
  - Content independence (unlike RPC approaches)
- Optional Transactional capability (2PC)
- Various Execution modes
- Desire for high throughput and low latency
  - optional data batching and command pipeline
  - Binary encoding key
- Security
- Traffic Sensitive Heartbeating
- Optional High Availability

Example Protocol Semantics

- ***Command <path to resource> [Optional Parameters]***
- GET /RIB/2/interfaces/1
  - Client gets the entry with ifindex 1 from the RIB resource on controller
- GET /RIB/1/interfaces
  - Client gets the whole interface table from the RIB instance 1
- DEL /RIB/1/Rib/routes
  - Flush every route in the RIB table
- SET /RIB/1/Rib/routes {route entry contents}
  - Client creates or updates the RIB on instance 1 with a new route
- REPORT /RIB/10/Rib/RouteAdded {route entry contents}
  - Manager reports to subscribed agents a newly added route

ForCES Architecture Gaps

- Directionality
  - ForCES assumes the Resource owner (RIB manager/agent) will associate with the Resource Controller (Client)
- Requires Protocol change to allow the reverse

ForCES Architecture Gaps

- Client Knowledge
  - ForCES assumes the Client is a sage
    - Knows everything and controls every resource
      - Not true in the case of RIB manager controlled by many clients
- Requires small protocol changes to accommodate for a slightly dumber controller client
  - New Path flags: Table CREATE, Exclusive CREATE and APPEND

ForCES Architecture Gaps

- Authentication and Authorization
  - Assumption of single resource control point
    - Resource owner doesnt need to know who they are
- Would TLS and certificates solve this?
  - Implementation of the RIB manager would keep track of the different identities

ForCES Architecture Gaps

- Multi headed control missing
  - Assumption of single resource owner bites again
- May require a protocol change that allows ownership of specific LFB instances or parts-of to be tagged by some owner ID
  - Was originally brought up as a requirement for scaling the Resource ownership but rejected for the new charter
    - http://www.ietf.org/proceedings/86/slides/slides-86-forces-7.pdf

ForCES Architecture Gaps

- RFC 5811 TML may not be a good fit for I2RS
  - May Require creating a new TML
- Not being RPC based may be negative
  - An RPC may result in one message whereas an equivalent transaction for ForCES would need to be broken down to multiple messages
    - Not an atomicity problem but usability issue
  - An RPC message may be more descriptive
    - Example: In ForCES to subscribe to an event, you SET a property of the Event

Pros/Cons

- Cons
  - Not RPC
  - Protocol Changes required to fully meet requirements

- Pros
  - Not RPC
  - Simple and extensible protocol
  - Designed for high throughput + low latency
  - Transport independence
  - Capability discovery/negotiation
  - Pub-subscribe events
  - Rich transactional features
  - High Availability