

Application Layer Multicast Extensions to RELOAD
draft-kolberg-sam-baseline-protocol-00

ALM Usage for RELOAD

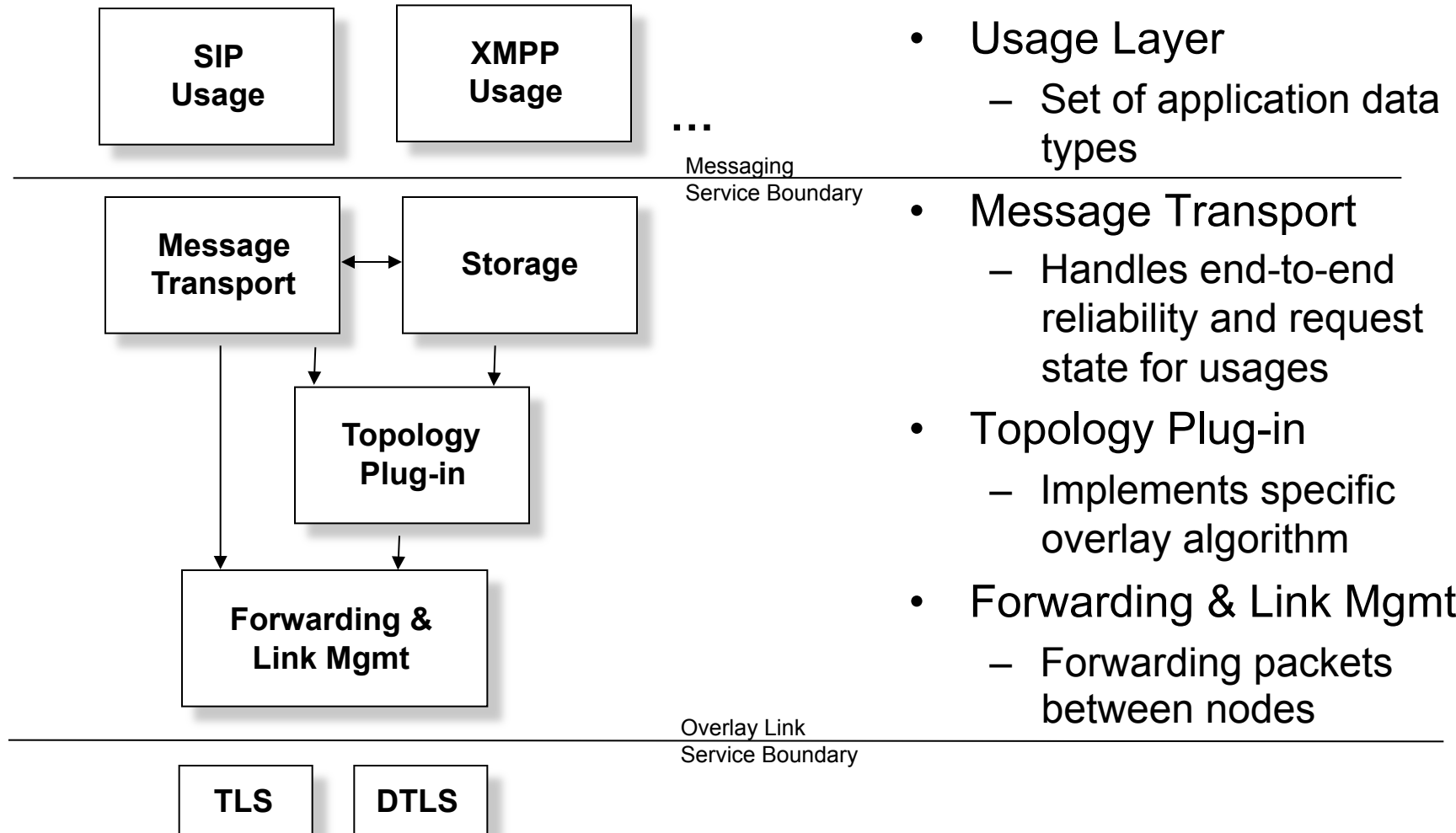
John Buford, Mario Kolberg

3/29/2011

Overview

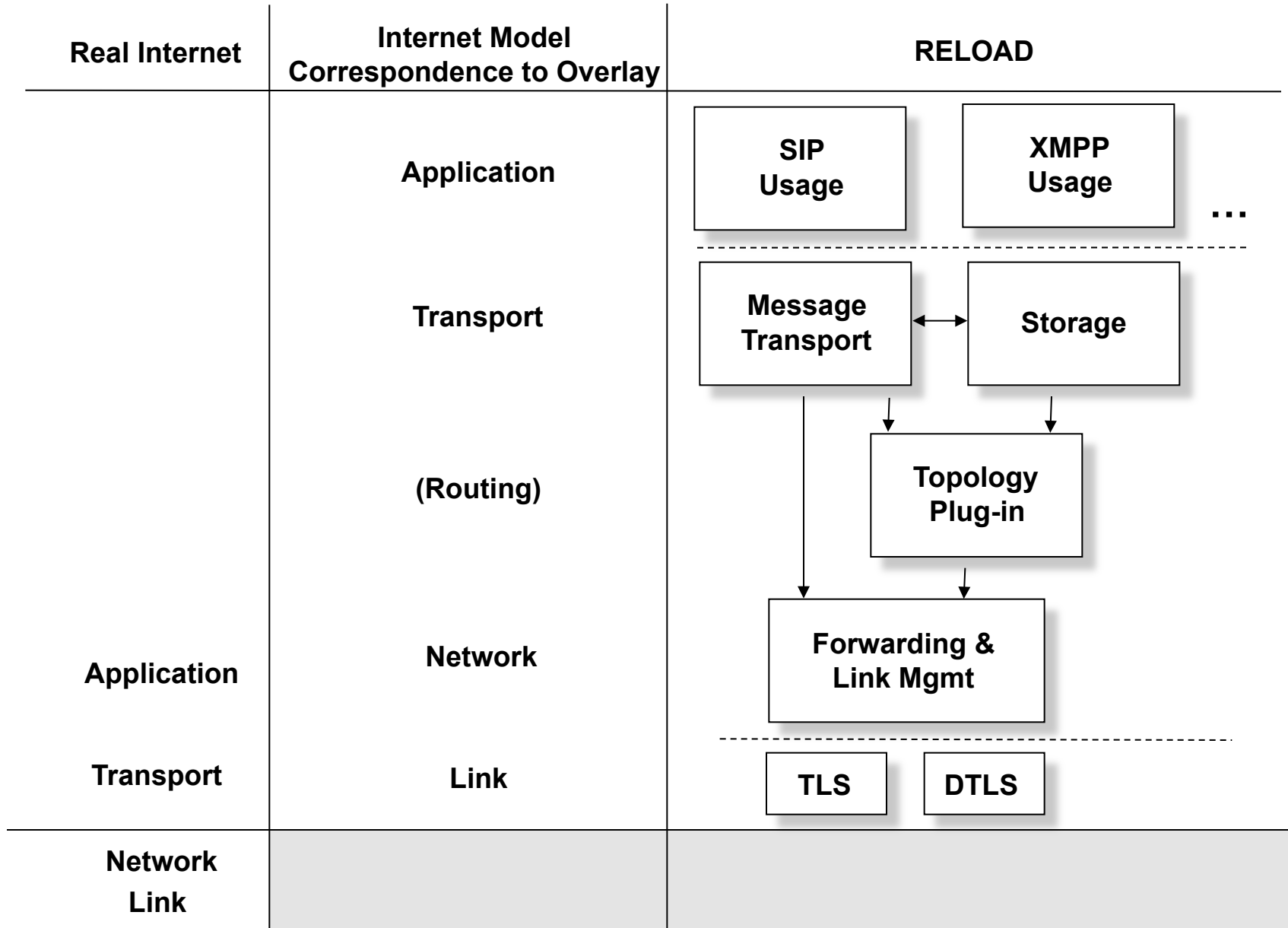
- Baseline document
 - Extension to RELOAD, uses P2PSIP RELOAD as the overlay layer
 - Split document:
 - This document concentrates on ALM only RELOAD Usage
 - Another document soon on Hybrid ALM RELOAD Usage
 - Initial focus for ALM only is using a Scribe like algorithm for ALM, more algorithms later
- Any comments welcome!

P2P-SIP Architecture

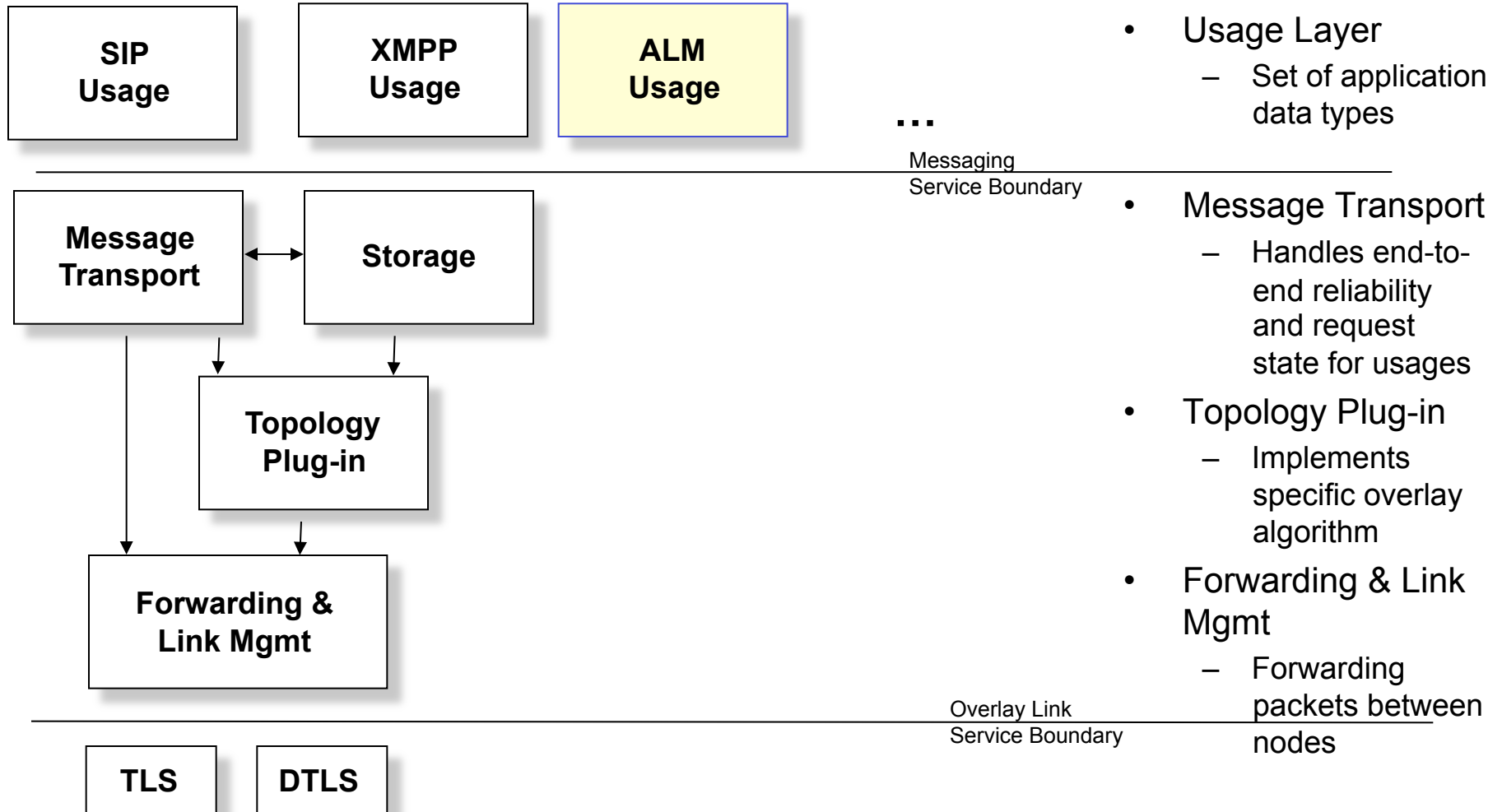


- Usage Layer
 - Set of application data types
- Message Transport
 - Handles end-to-end reliability and request state for usages
- Topology Plug-in
 - Implements specific overlay algorithm
- Forwarding & Link Mgmt
 - Forwarding packets between nodes

P2P-SIP Architecture

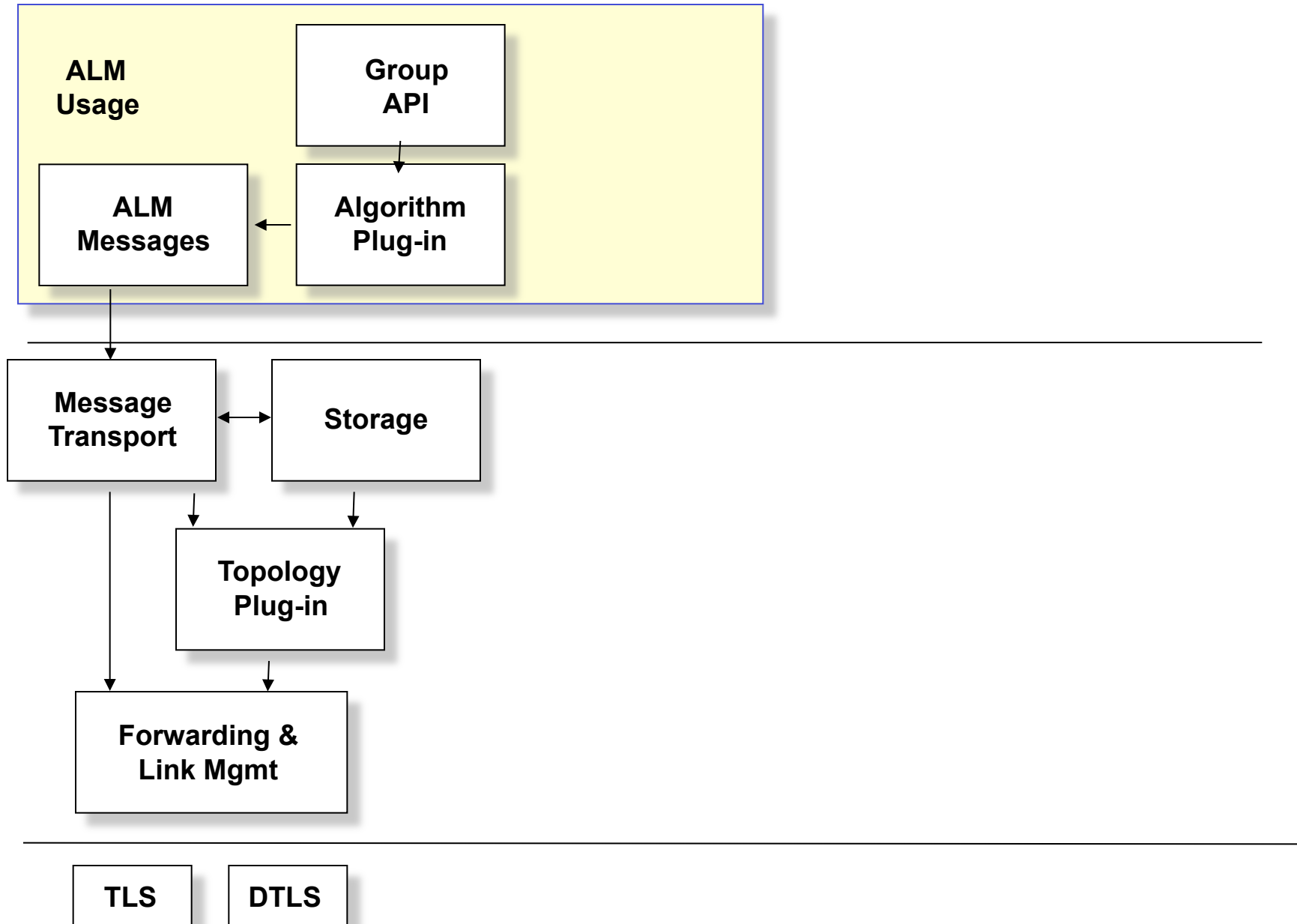


ALM Usage for RELOAD

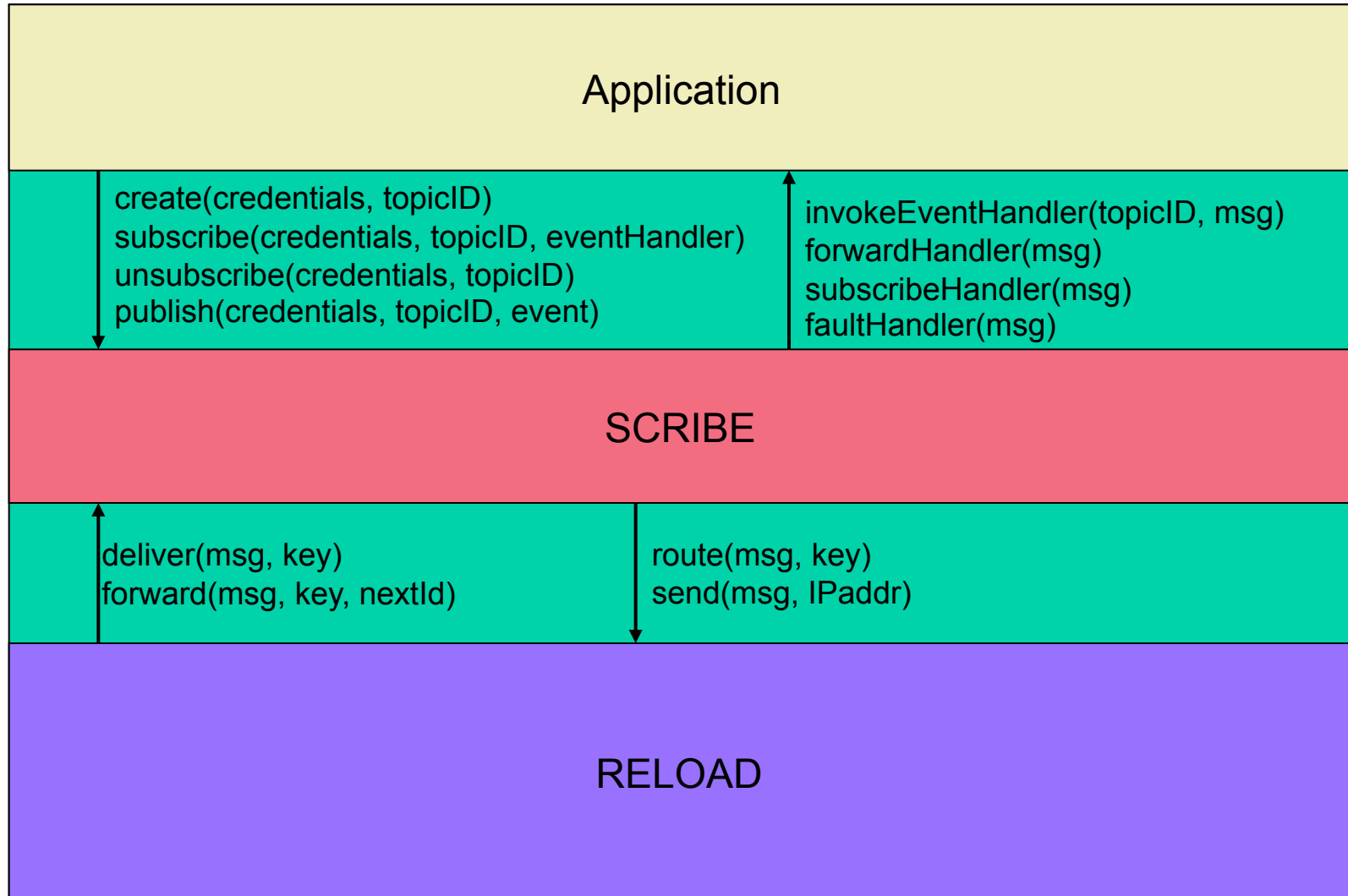


- Usage Layer
 - Set of application data types
- Message Transport
 - Handles end-to-end reliability and request state for usages
- Topology Plug-in
 - Implements specific overlay algorithm
- Forwarding & Link Mgmt
 - Forwarding packets between nodes

ALM Usage for RELOAD



Scribe



Message Mapping

Section of draft 01	RELOAD ALM Usage	Scribe [ref]
6.2.1	CreateALMTree	Create
6.2.2	Join	Join
6.2.3	JoinAccept	
6.2.4	JoinConfirm	
6.2.5	JoinDecline	
6.2.8	Leave	Leave
6.2.10	Reform	
6.2.11	Heartbeat	
new	Push/Deliver/Send	MULTICAST
		deliver(msg,key)
		forward(msg,key,nextID)
		route(msg,
		send(msg,IPAddr)

Scribe Pseudocode

```
CREATE : groups.add(msg.GroupId)
```

```
MULTICAST : foreach (groups[msg.GroupId].children as NodeId)  
             SEND(msg, NodeId)  
             if memberOf(msg.GroupId)  
                 invokeMessageHandler(msg.GroupId, msg)
```

```
LEAVE : groups[msg.GroupId].children.remove(msg.source)  
        if (groups[msg.group].children = 0)  
            SEND(msg, groups[msg.GroupId].parent)
```

Scribe Pseudocode cont.

```
JOIN : if (checkAccept (msg) ) {
        recvJoins.add (msg.source, msgGroupId)
        SEND (JOINAccept (nodeID, msg.source, msg.GroupId) )
    }

JOINDecline: if (recvJoins.contains (msg.source, msg.GroupId) )
              recvJoins.del (msg.source, msgGroupId)

JOINConfirm: if (recvJoins.contains (msg.source, msg.GroupId) ) {
              if !(groups.contains (msg.GroupId) ) {
                  groups.add (msg.GroupId)
                  SEND (msg, msg.GroupId)
              }
              groups [msg.GroupId] .children.add (msg.source)
              recvJoins.del (msg.source, msgGroupId)
          }
```

Next Steps

- Update draft on ALM Usage for RELOAD
- Start working on the Hybrid ALM Usage for RELOAD