

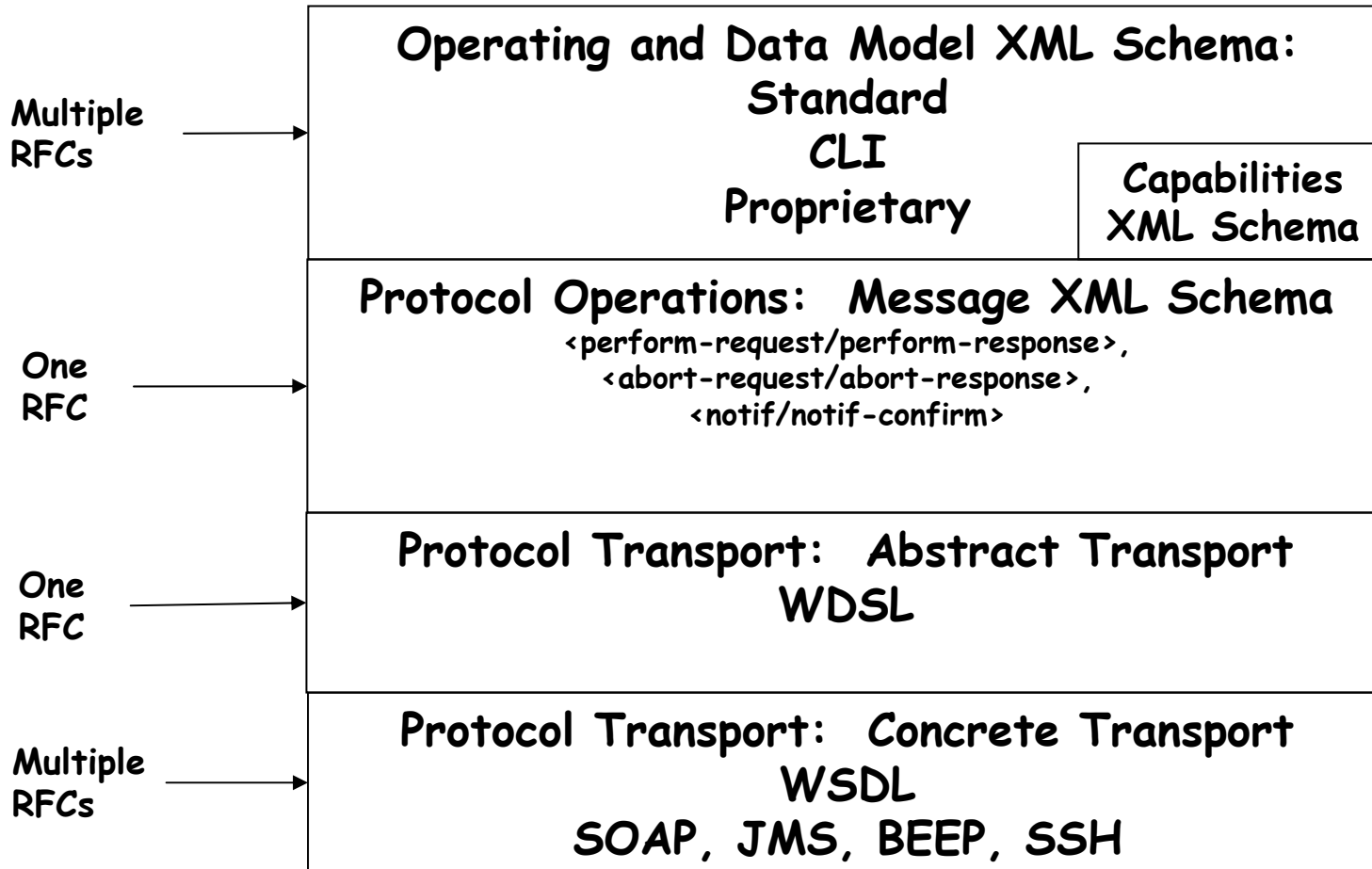


XML Network Management Interface (draft-weijing-netconf-interface-00.txt)

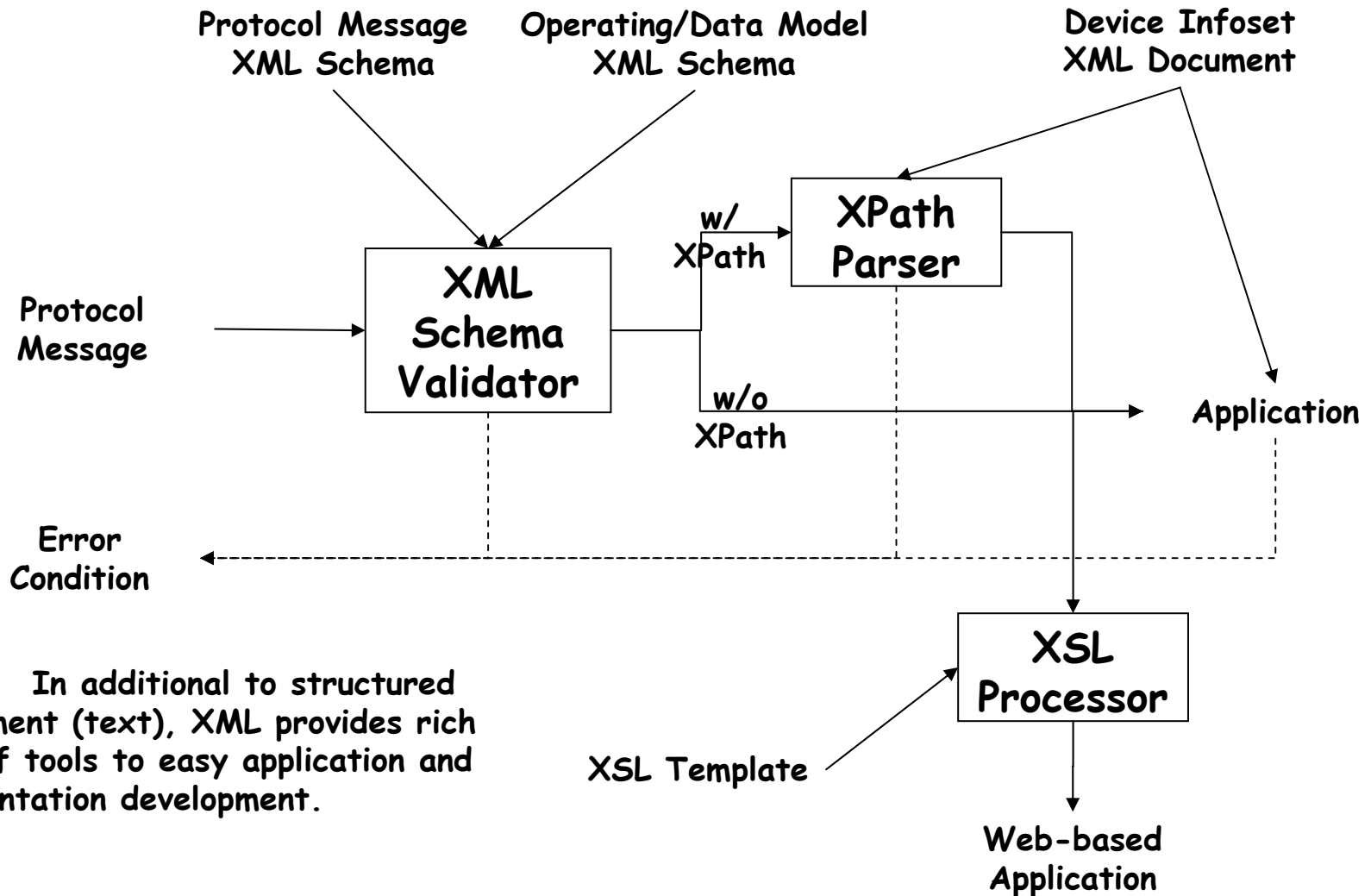
57th IETF

Weijing Chen
wchen@labs.sbc.com
Keith Allen
kallen@labs.sbc.com

Interface Components:

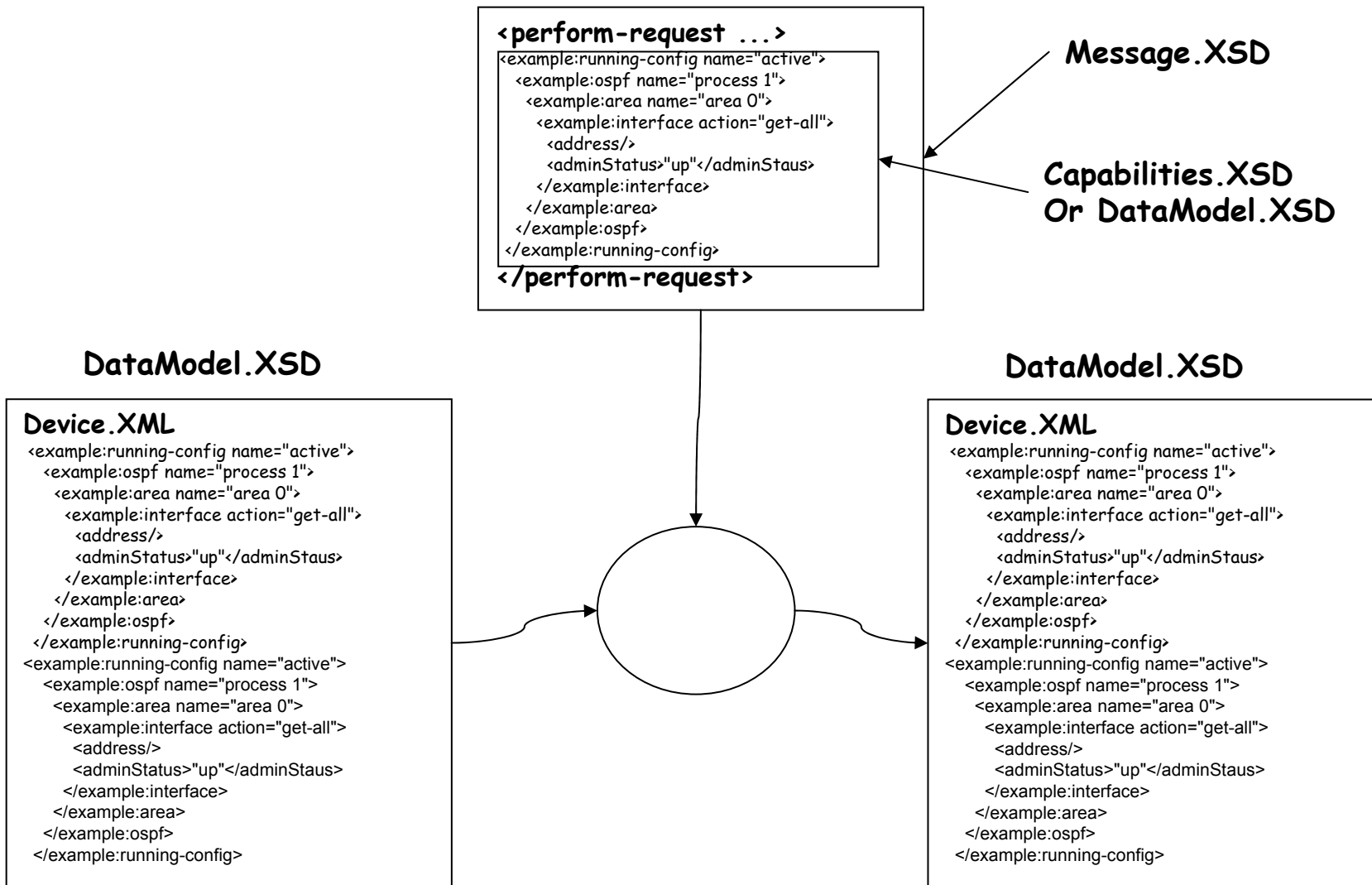


Processing Flow:



Note: In addition to structured document (text), XML provides rich set of tools to easy application and presentation development.

XML Schema and Document:



Capabilities Schema

- **The capabilities of optional operation function are described in capabilities XML schema. Very simple devices will support only the minimum, while more complex devices will be expected to support more.**
 - Transaction
 - Notification
 - Data model schema
- **It allows peers to exchange the actually functionality implemented in other end using regular protocol message, no special message required.**
 - perform-request/performance-response
- **It is really a standardized data model schema, the first try of defining a standard data model.**

Operating Model Schema

- **Again, different devices will support various management operating models. The proper operating model would be described in a separate operating model schema or a separate part of data model schema.**
 - **Candidate, running, startup config**
 - **Validate, copy-config**
 - **Lock, unlock, commit, rollback**
 - **Multiple channels**
 - **Kill a session**
- **Section 4 of draft-weijing is a rough description of such operating model schema.**
- **We feel that the management operations should be as simple and flexible as possible, with the operating and data model being the focus for defining the complexity of the device.**
 - **Separating the operating model from the protocol message work would also help to increase the WG focus.**

Configuration vs. state data

- **First, operation-as-attribute is to address the ambiguity problem on overwrites that has been discussed in the list. The draft-enns-01 also takes similar approach.**
- **Second, it is to address configuration vs. state data:**
 - **The idea of being able to retrieve just the configurable parameters of a system, so that they may be stored is very appealing.**
 - **Two possible mechanisms to distinguish configuration data vs. state data:**
 - **Place “configuration” and “state” in separate subtrees (separate schema or separate parts of the schema). We're afraid it will become overly tedious to define, use, and maintain separate element trees for state and config data (e.g. two “interface” elements under two trees).**
 - **Via “configuration (readwrite)” or “state (readonly)” attributes. For example, “administrative state” of the interface would be defined with “readwrite” attribute (operation-as-attribute) while “operational state” would be defined with “readonly” attribute. Thus, a “get-readwrite” operation on an interface element would include “administrative state” in its return, while a “get-readonly” operation would include “operational state.” It's sort of a rudimental filtering capability.**

WSDL

- **The use of WSDL in this draft implied that WSDL is generic enough to describe the SSH and BEEP bindings as well as the SOAP binding. But neither BEEP nor SSH are currently included in the WSDL specification, and neither BEEP nor SSH is currently supported by WSDL tools.**
- **With some efforts, we will be able to describe SSH and BEEP bindings in WSDL specification.**
- **But the SSH and BEEP WSDL tools must be developed by tool vendors.**