

Evolution of the Subscription & Event Notification Drafts

IETF #98 Chicago

Eric Voit

28-Mar-2017

DRAFT

Authors on at least 1 drafts

Andy Bierman

Alexander Clemm

Balazs Lengyel

Einar Nilsen-Nygaard

Alberto Gonzalez Prieto

Ambika Prasad Tripathy

Eric Voit

+ Contributors

Sharon Chisholm

Yan Gang

Peipei Guo

Susan Hares

Tim Jenkins

Michael Scharf

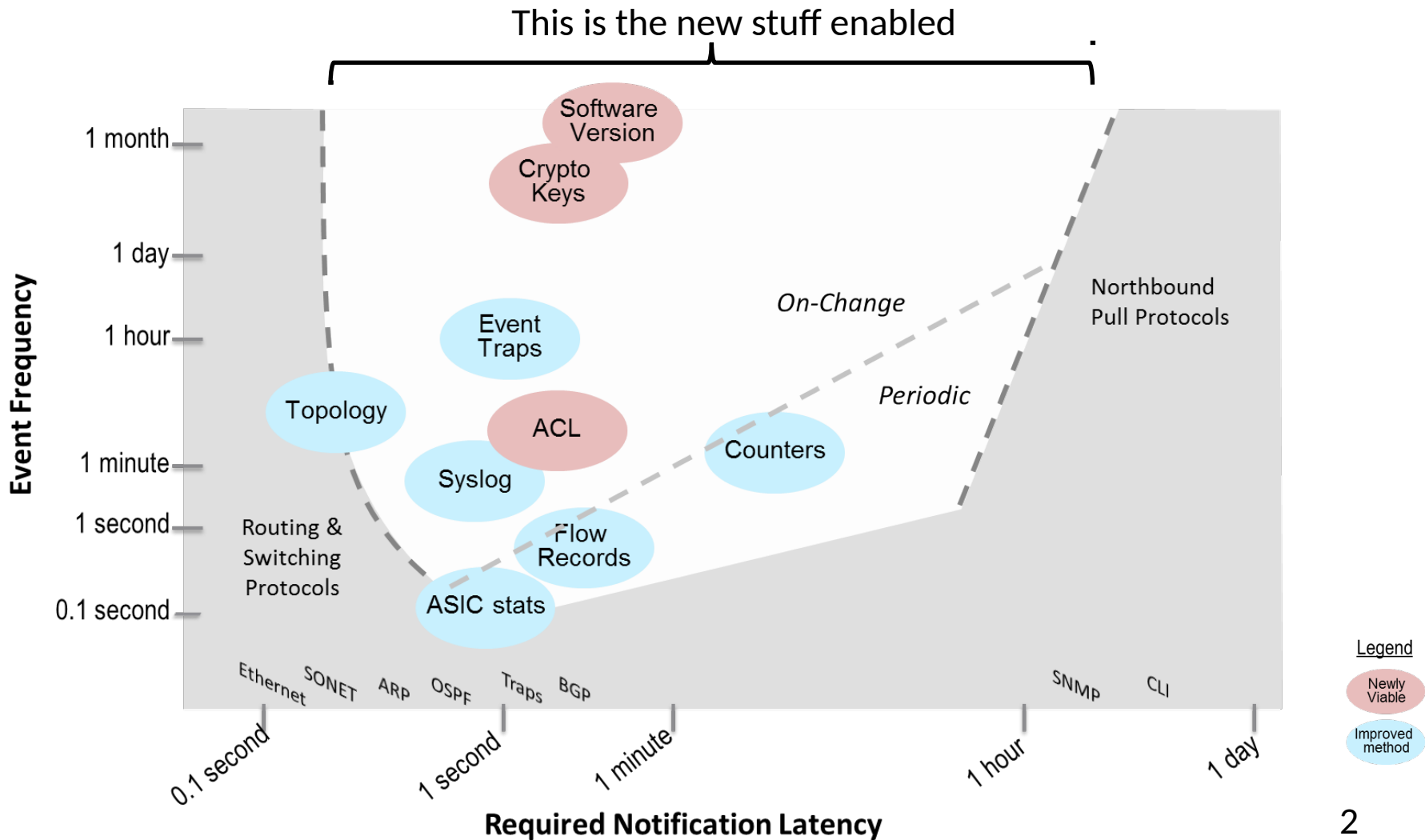
Hector Trevino

Kent Watsen

Guangying Zheng (Walker)

The now successfully
retired[™]
Dezign™ Team

Data Replication Frequency vs. Latency



Functional Partitioning Summary

		Subscribed Notifications	YANG Datastore Push
Subscription	Types	Dynamic and Configured	
	per Session	many	
	Negotiation	Yes	
	RPCs	establish, modify, delete, kill	
	State Notifications	started, suspended, resumed, terminated, modified	
Data Plane Notifications	basic	RFC-7950 + Subscription-ID	complete and changes
	Headers	Yes	
	Update Bundling	Yes	
Transport	NETCONF	Yes	
	RESTConf, HTTP, HTTP2	Yes	

Legend

draft-ietf-netconf-yang-push
draft-ietf-netconf-subscribed-notifications
draft-voit-netmod-yang-notifications2
draft-ietf-netconf-event-netconf
draft-ietf-netconf-event-restconf

Functionality per Draft

Subscribed Notifications

- Dynamic & Configured subscriptions
- Multiple subscriptions / transport
- Multiple configured receivers
- Establish, modify, delete, kill RPC
- State change notifications
- Suspend/resume
- Filtering full notifications
- Stream discovery
- Replay (and start time negotiation)
- Prioritization
- Monitoring / reporting
- QoS
- Error responses

YANG Datastore Push

- Datastore on-change and periodic triggers
- Filtering objects within a notification
- Authorization model per object
- Sending of full YANG trees or yang-patch
- Tagging of partial updates
- Tagging of on-change object support
- Negotiation of filters and period lengths
- More error responses

YANG Notifications2

- Encapsulation Headers objects: Signature, de-duplication, severity, originator
- Bundled records and record types

NETCONF Transport for Subscribed Notifications

- Transport mapping

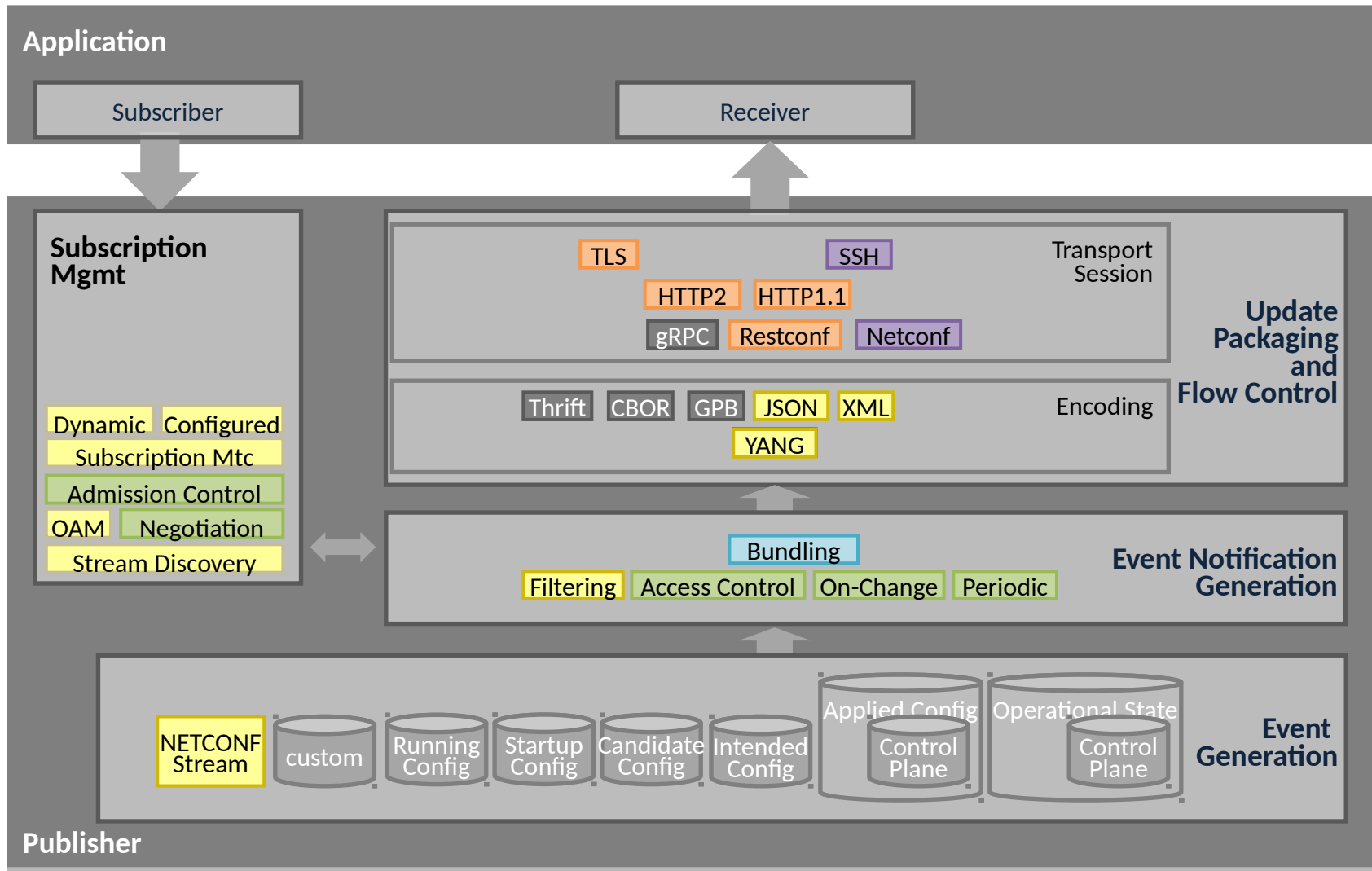
RESTCONF & HTTP2 Transport for Subscribed Notifications

- Transport mappings (including HTTP2 QoS)
- Heartbeats and clean-up

Legend

draft-ietf-netconf-yang-push
draft-ietf-netconf-subscribed-notifications
draft-voit-netmod-yang-notifications2
draft-ietf-netconf-event-netconf
draft-ietf-netconf-event-restconf

Drafts in Layered Framework



Legend

draft-ietf-netconf-yang-push
draft-ietf-netconf-subscribed-notifications
draft-voit-netmod-yang-notifications2
draft-ietf-netconf-event-netconf
draft-ietf-netconf-event-restconf

yang push

Updates since IETF #97

- -05 revision
- Ability to get operational data from filters
- Extension notifiable-on-change added
- New appendix on potential futures
- New error and hint mechanisms included in text and in the YANG model
- Updated examples based on the error definitions
- Text updates

yang push

Final steps before WG Last Call

- Subscription ID as an identifier only relevant to a single receiver
- Deferral of a standard header and bundle support (i.e., use the current data plane notifications.)

subscribed-notifications

Updates since IETF #97

- Move away from 5277bis
- Kill subscription RPC added
- Error conditions added
- YANG model simplifications
- Renaming of Subscription State Notifications and identifiers

subscribed-notifications

Final steps before WG Last Call (Same as for YANG-Push)

- Subscription ID as an identifier only relevant to a single receiver
- Deferral of a standard header and bundle support (i.e., use the current data plane notifications.)

Updates since IETF #97

- -02 revision
- Removed sections redundant with other drafts
- 3rd party subscriptions out of scope
- SSE only used with RESTCONF and HTTP1.1 Dynamic Subscriptions.

Final steps before WG Last Call

HTTP2 transport message compatibility with GRPC

- One set of meetings. Need another set of eyes

Notif-netconf

No Updates since IETF #97

- -02 revision coming shortly
(was awaiting 5277bis Charter solidification)

notifications2

draft-voit-netmod-yang-notifications2 Proposes Solutions to the Following:

1. What are the set of transport agnostic header objects which might be usefully placed within YANG notifications?
2. How might a set of YANG notifications be bundled into a single transport message?
3. How do you query the originator of a notification to troubleshoot the bundling process?

notifications2

#1 Transport Agnostic Header Objects

```
+---n notification-message
  +--ro notification-message-header
  | +--ro record-time
  | +--ro record-type?
  | +--ro record-id?
  | +--ro record-severity?
  | +--ro observation-domain-id?
  | +--ro subscription-id?
  | +--ro notification-time?
  | +--ro notification-id?
  | +--ro previous-notification-id?
  | +--ro signature?
  | +--ro message-generator-id?
  +--ro receiver-record-contents?
```

notifications2

#2 bundling multiple notifications into a single transportable message

```
+---n bundled-notification-message
  +--ro notification-message-header
  |   +--ro notification-time
  |   +--ro notification-id?
  |   +--ro previous-notification-id?
  |   +--ro signature?
  |   +--ro message-generator-id?
  |   +--ro record-count?
  +--ro notification-records*
    +--ro notification-record-header
    |   +--ro record-time
    |   +--ro record-type?
    |   +--ro record-id?
    |   +--ro record-severity?
    |   +--ro observation-domain-id?
    |   +--ro subscription-id?
    +--ro receiver-record-contents?
```

Thank you!