Push It

Open Issues

Core Issues

Aggregation / Registration

Subscription Split

Payload Formats

Encryption

Time To Live

Delivery Receipts / Acknowledgment

High Order Bit

For many of these, we first need to decide whether a solution is needed now (core)

...or maybe later (extension)

The Other High Order Bit

We’re here to make decisions*

- No outcome here doesn’t mean we’ve failed
- ...but…
- Maybe we can at least plan for a decision
- What do you/we need to decide

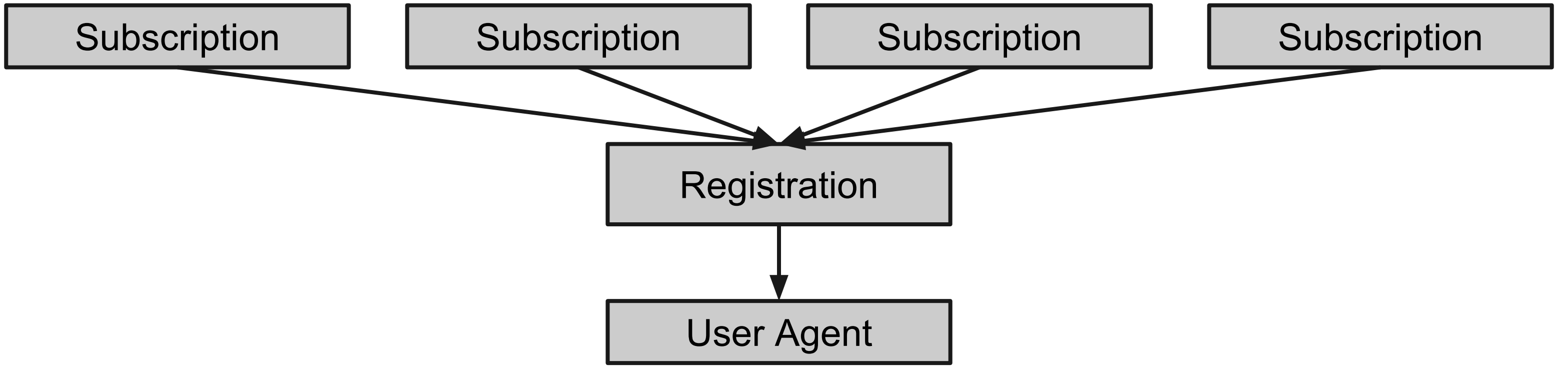
*Decisions that will be confirmed on the mailing list, of course.

Aggregation Option

User agents create a registration

Subscriptions are associated with a registration

A single resource is used to monitor all subscriptions for a registration

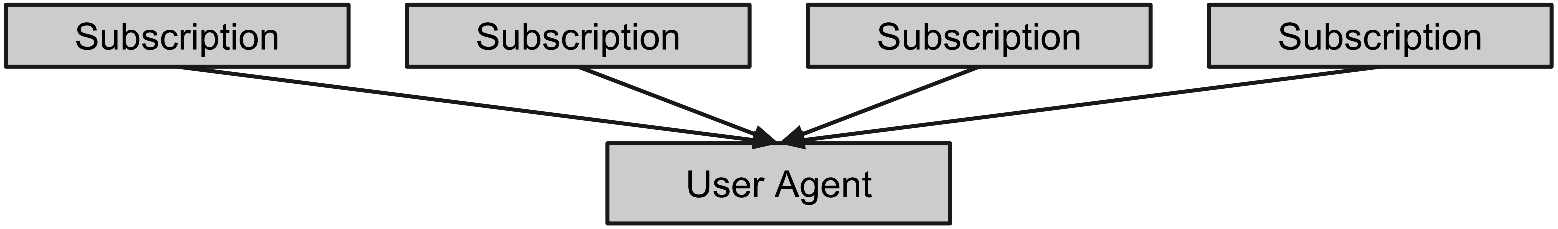


Disaggregation Option

User agents only create subscriptions

Each subscription is monitored independently

Rely on HTTP/2 multiplexing to put all monitoring on the same connection.



Aggregate?

Aggregation

Single action for getting all push messages. Easier to repair, easier to create subscriptions. More efficient to start up.

Aggregation happens anyway at the connection.

Disaggregation

Push service can disaggregate subscriptions. UA maintains state, the push service doesn’t need to search for active subscriptions when a UA connects.

No registration step.

I need more information

Subscription Split

Proposal to have user agents and application servers see different views on a subscription

Pros

- Makes privilege separation easier (e.g. deleting a subscription is user agent only)

Cons

- More resources, PUT becomes opaque to HTTP

- List preference seemed to be for a split.

Payload Format

Lots of discussion about defining some sort of JSON format on messages to support various features

Proposal: defer this until we’ve talked about encryption, plus some of those features.

Encryption

Overview given at last meeting

Goal: provide end-to-end confidentiality and integrity for push messages

We seem to have consensus to do it

Format choice is the only issue

Proposal: defer for now

TTL

How long does a server retain a push message?

Can applications request a maximum TTL?

Do we need to fix a minimum TTL?

...or, can we negotiate a value?

Request a minimum

Application signals TTL with a push request
Relative time preferred (see list discussion)

Choices are yes, later or separate doc, or no

How to signal is secondary
...options are Timeout header, new header, extended payload format