

WebPush + HTTP/2 + IoT

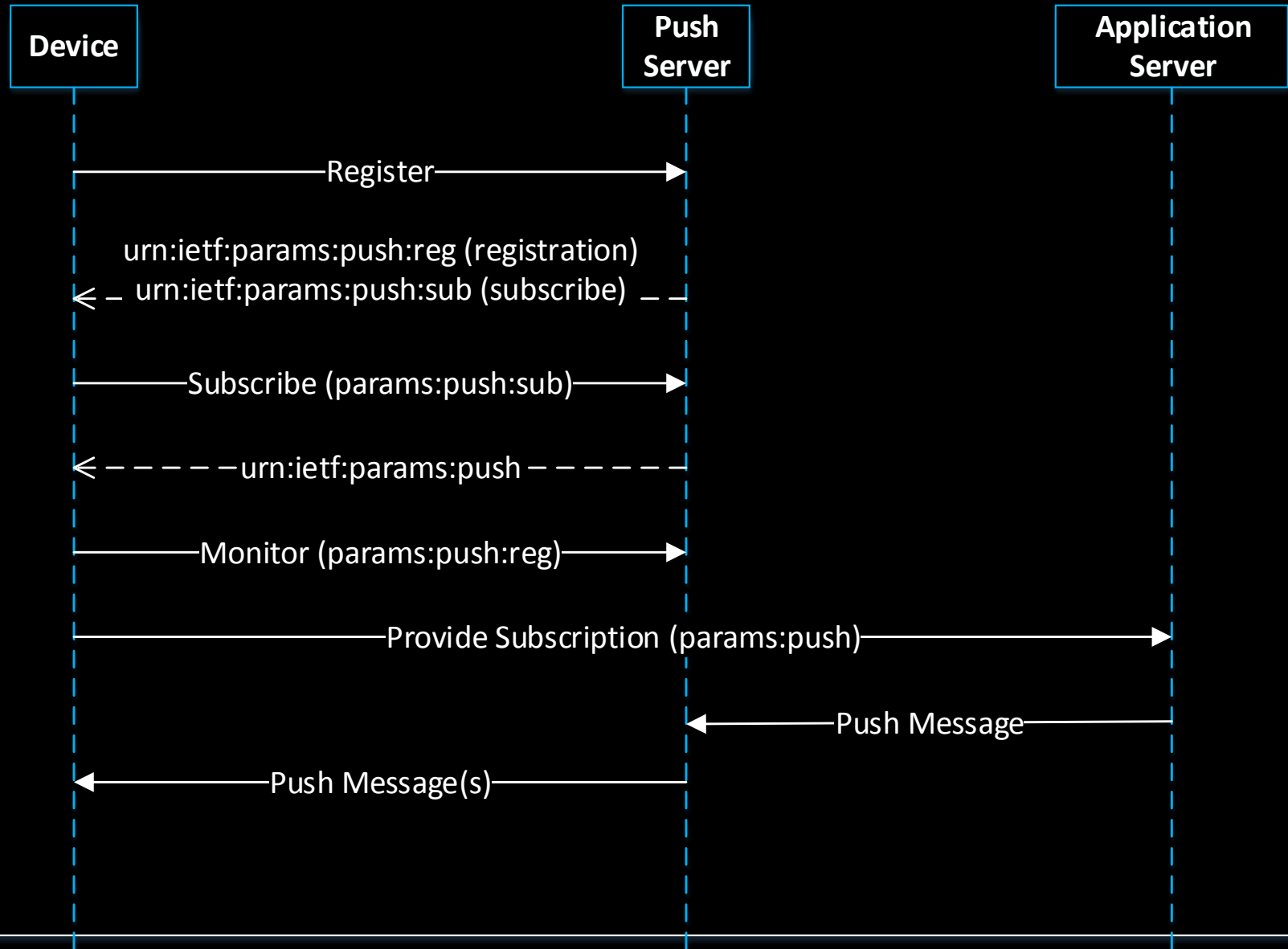
IETF 92

Push Scenarios

There are two primary scenarios which must be addressed:

- Web applications in a mobile user agent and
- Embedded devices receiving push messages from cloud services through an intermediate *field gateway* - a reasonably powerful device (capable of secure HTTP/2 communications), which acts as a local agent.

draft-thomson (Simplified Flow)



Feedback: Registration

To enable aggregation of push message delivery, Registration adds complexity to the protocol (both the registration and expiration process) and requires the Push Server to manage the overhead of Registration-Subscription relationships.

There are scenarios where aggregation is not required.

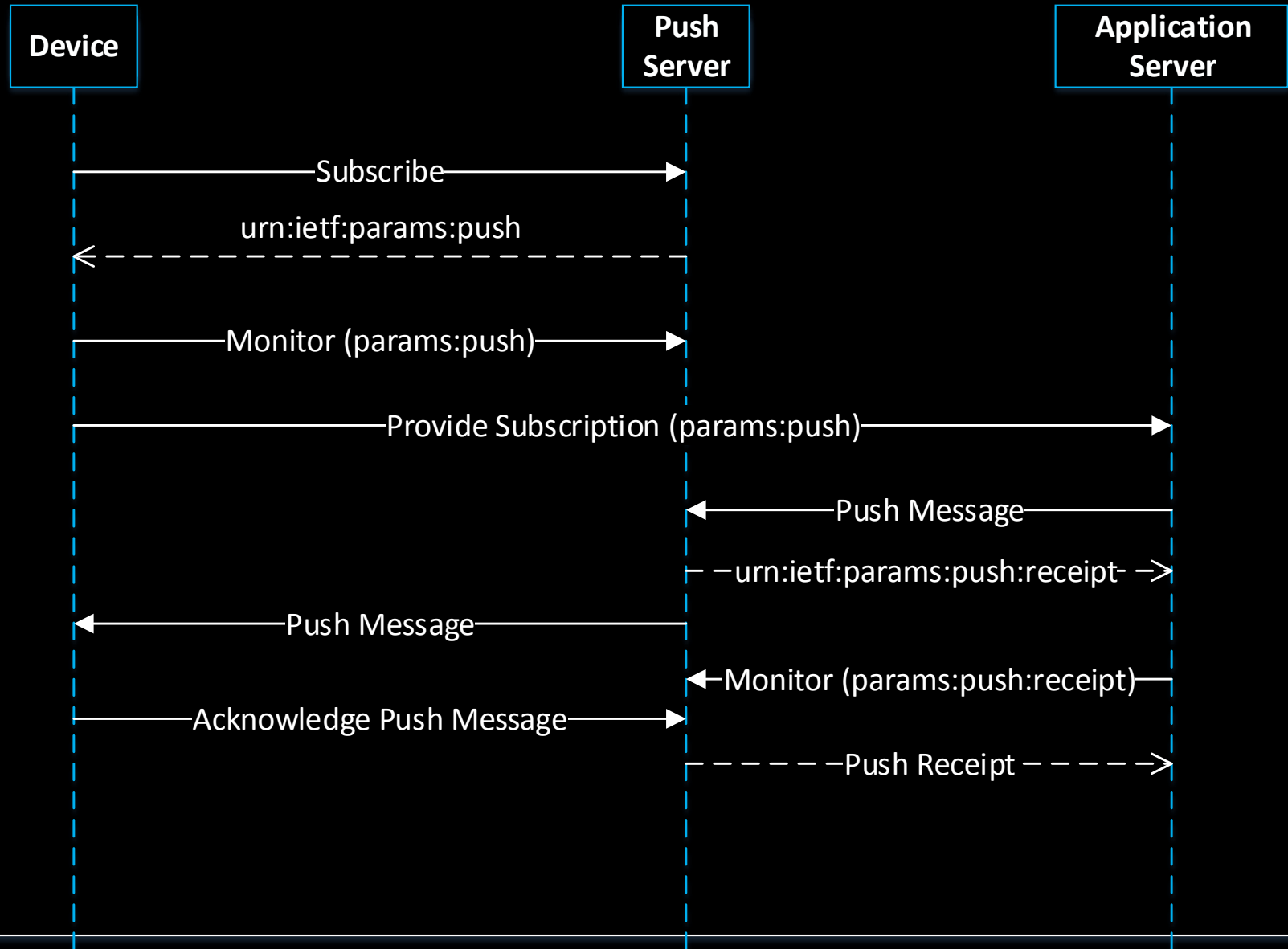
Feedback: Message Expiration

A push server that does not allow the application server to recommend an expiration (time-to-live) for a message may store and deliver messages that are stale, incurring additional power drain on the device for a message to be discarded.

Feedback: Application Reliability

A push server that does not support reliable delivery over intermittent network connections or failing applications on devices, forces the device to acknowledge receipt directly to the application server, incurring additional power drain in order to establish (usually secure) connections to the individual application servers.

draft-damaggio (Simplified Flow)



Proposed Changes to draft-thomson

- Remove dependency on Registration/Subscribe resources
 - Simplifies protocol flow and subscription lifecycle
 - Simplifies the Push Server by eliminating maintenance of Registration-Subscription relationships
- Provide two methods for the User Agent to request push messages:
 - Request on Subscription resource
 - Request on Subscription resource (w/Prefer Wait)
- Eliminate aggregation (performing a GET on a Registration resource to receive messages for all subscriptions for a User Agent)
- Eliminate collapsing (performing a GET on a Subscription resource to get the last message).
 - Different collapsing or coalescing policies for messages can be added if required.

Proposed Changes to draft-thomson

- Add a requirement for the User Agent to confirm delivery of messages from the Push Server
- Add a request message format (JSON) for push messages
 - Include *request_receipt* option to allow the Application Server to request a delivery confirmation from the Push Server
 - Include *time_to_live* option to allow Application Server to recommend a message expiration to the Push Server

Open Issues

- The response message format of the delivery receipt is TBD.
- Should the push server return a 400 if the requested `time_to_live` exceeds its storage limits?
- How would proposed message encryption impact Push Server *directives* such as such as *request_receipt* and *time_to_live*?
 - Should directives be modeled as custom HTTP headers?

Next Steps

We would like to explore whether a mind-meld of draft-thomson and draft-damaggio could be the basis for a WebPush-00 draft.