Secure Dimensions GmbH

Holistic Geosecurity

Towards a generic OGC API Async Recommendation

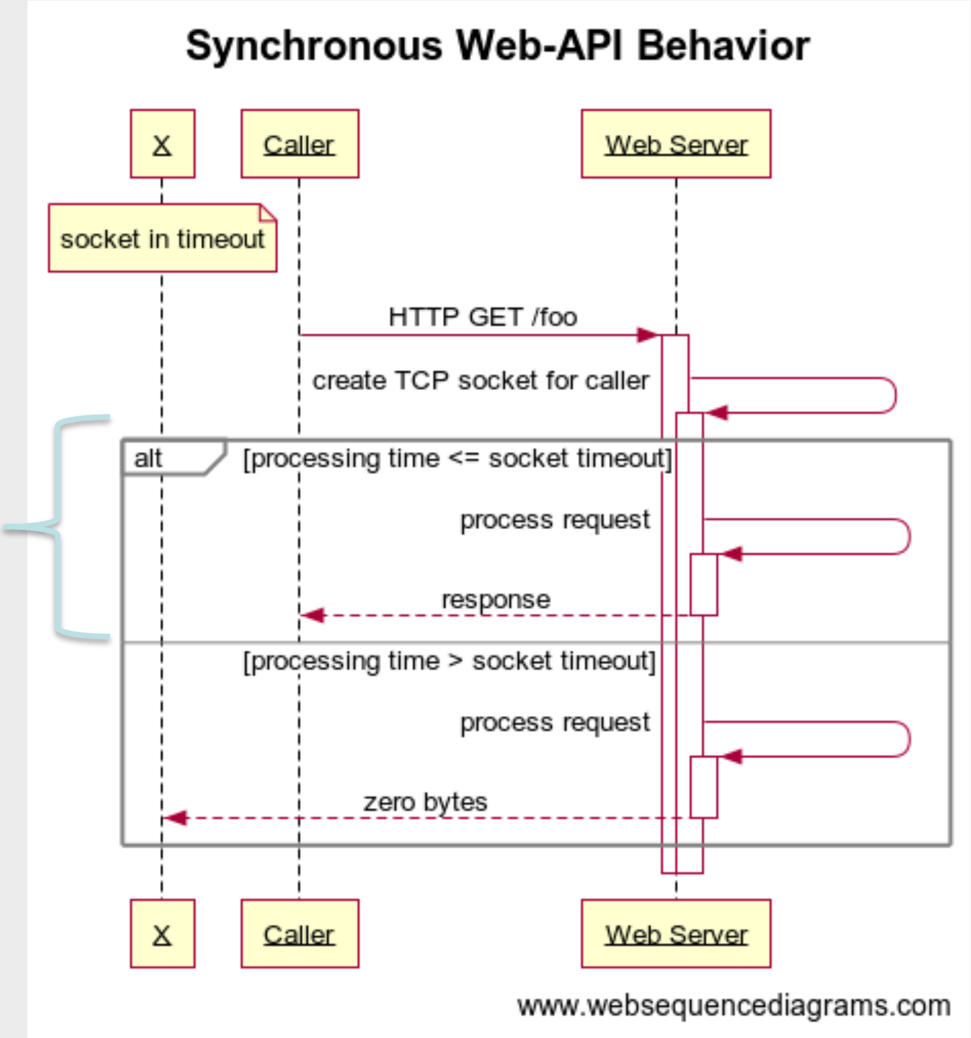
OGC Code Sprint
London
14 – 16. September 2022

Andreas Matheus
Secure Dimensions GmbH

Synchronous Web-API Behavior

Problem

OK


Synchronous Web-API Behavior
X
Caller
Web Server
socket in timeout
HTTP GET /foo
create TCP socket for caller
alt
[processing time <= socket timeout]
process request
response
[processing time > socket timeout]
process request
zero bytes
X
Caller
Web Server
www.websequencediagrams.com

Synchronous Web-API
-- Technical Restrictions --

- Send and receive in same TCP socket pair

⇒Consequence: Response is due BEFORE socket timeout happens

⇒Caller can set socket timeout, but this has no effect in Reverse-Proxy setup (e.g. Nginx + upstream)

⇒The response must be send back BEFORE the default socket latency is reached! – typically 30s

⇒But users won’t even wait that long. So response should come back within approx. 10s!

Critical example of OGC API where Synchronous may not work!

- From Testbed 17: (Encrypted) GeoPackage with large Feature Collection
- GeoPackage aka SQLite is a binary file that must be fully created on the server side BEFORE the first byte of the response can be sent back to the caller

⇒Danger that socket timeout happens BEFORE GeoPackage file is created!

⇒Danger that server resources (involved in creating the GeoPackage) are wasted

⇒Danger of DoS as repeated requests may consume all available server resources!

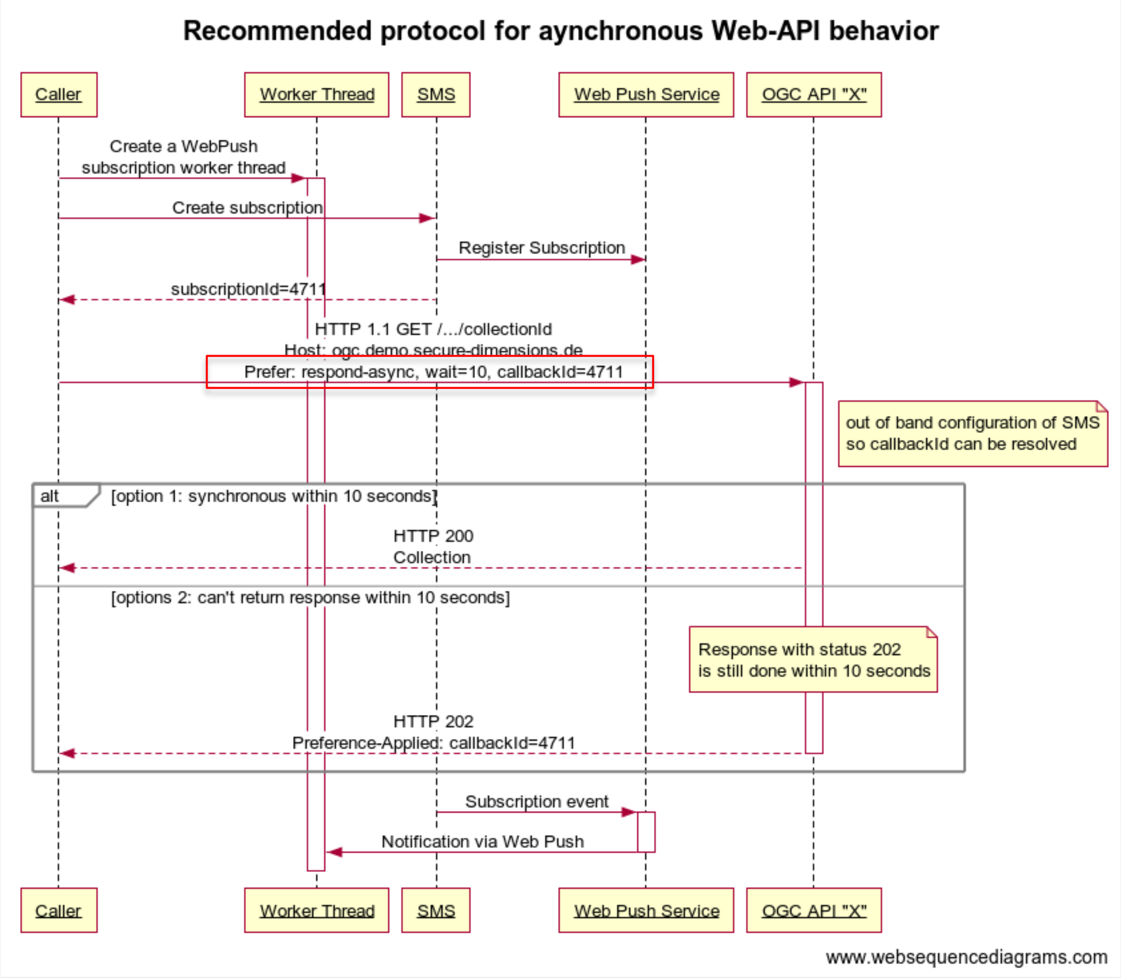
Asynchronous Web-API Behavior

- Ensure that SOME response is send back to the caller BEFORE the socket timeout happens!
- The continuation protocol typically depends on the API's semantics. Two basic variations exist:
  - The caller continuous to be the active part (polling)
    - E.g. 303 "See Other" with a URI to a status monitor to be used by the caller to get more information. Protocol and data structure typically not standardized.
  - The service is the active part (push)
    - E.g. 202 "Don't call us – we'll call you"
    - => But how to tell the client what will happen next?

Testbed 18 Recommendation (Secure and Asynchronous Catalogue Task)

- Caller shall use “Prefer” to indicate the preference for the response
  - RFC 7240 semantics (wait for synchronous response or “I am able to handle async continuation)
- Testbed 18 proposal is that the HTTP header includes a “callback” or “callbackId”
  - Prefer: respond-async, wait=10, callback=https://ogc.demo.secure-dimensions.de/sms/subscriptions/4711
  - Prefer: respond-async, wait=10, callbackId=4711
- More details in OAB issue 1162

Sequence Example for Web Push



Recommended protocol for aynchronous Web-API behavior

Caller

Worker Thread

SMS

Web Push Service

OGC API "X"

Create a WebPush
subscription worker thread

Create subscription

Register Subscription

subscriptionId=4711

HTTP 1.1 GET /.../collectionId
Host: ogc.demo.secure-dimensions.de
Prefer: respond-async, wait=10, callbackId=4711

out of band configuration of SMS
so callbackId can be resolved

alt [option 1: synchronous within 10 seconds]

HTTP 200
Collection

[options 2: can't return response within 10 seconds]

Response with status 202
is still done within 10 seconds

HTTP 202
Preference-Applied: callbackId=4711

Subscription event

Notification via Web Push

Caller

Worker Thread

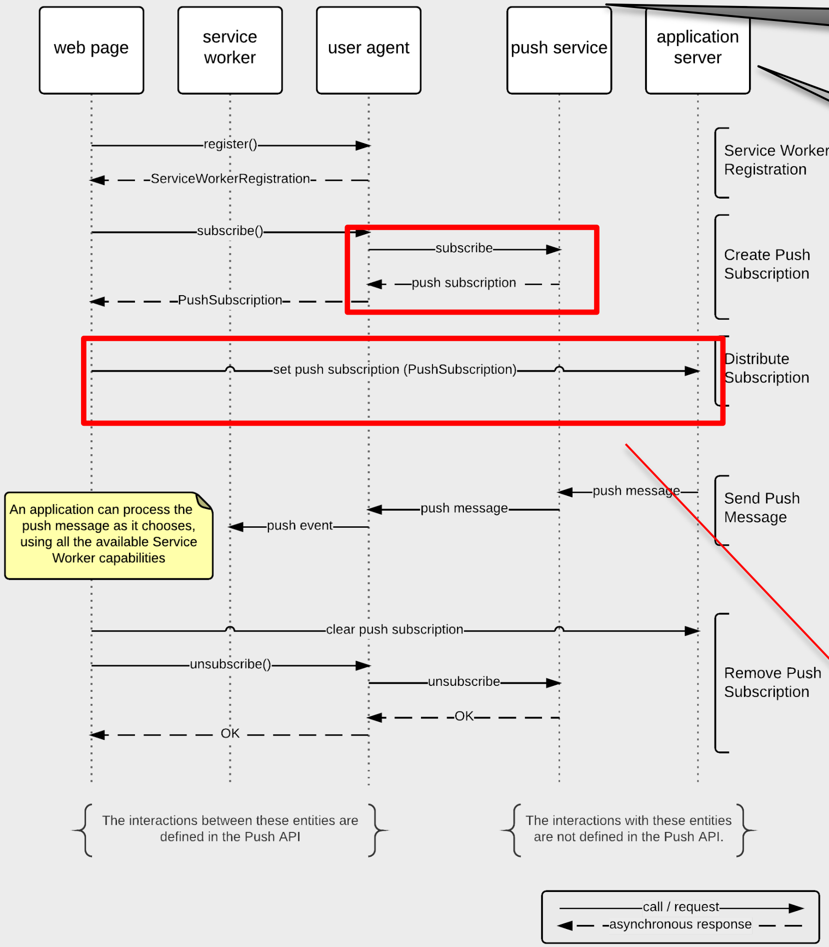
SMS

Web Push Service

OGC API "X"

www.websequencediagrams.com

Proposed Contribution during Code Sprint: Web Push


web page
service worker
user agent
push service
application server
register()
ServiceWorkerRegistration
Service Worker Registration
subscribe()
subscribe
push subscription
PushSubscription
Create Push Subscription
set push subscription (PushSubscription)
Distribute Subscription
push message
push message
push event
Send Push Message
An application can process the push message as it chooses, using all the available Service Worker capabilities
clear push subscription
unsubscribe()
unsubscribe
OK
OK
Remove Push Subscription
The interactions between these entities are defined in the Push API
The interactions with these entities are not defined in the Push API.
call / request
asynchronous response

Source: https://www.w3.org/TR/push-api/#sequence-diagram

TB18 SMS

TB18 pyCSW

pyCSW: https://ogc.demo.secure-dimensions.de/pycsw

SMS: https://ogc.demo.secure-dimensions.de/sms/api

To be implemented during code sprint

© 2022 Secure Dimensions

WebPush Notifications require Security ☺

- Voluntary Application Server Identification (VAPID) for Web Push
  - https://datatracker.ietf.org/doc/html/rfc8292
- Security Considerations
  - https://datatracker.ietf.org/doc/html/rfc8292#section-5
  - https://www.w3.org/TR/push-api/#security-and-privacy-considerations
- Pushed content should be …
  - authentic to the user => signature
  - confidential to the user => end-to-end encryption

Additional Use Case for Subscription Service

- You want to be informed if information changes on the service
  - Catalogue example:
    - Notify me if a particular record changes
    - Notify me if a record is published that I am looking for
- Three different kinds of “callback”
  - Via Email -> To user’s inbox – only a link
  - Via WebHoock -> To processing endpoint – can receive full resource(s) representations
  - Via WebPush -> To user’s browser / device – link to obtain resource(s)

Mission!

- Generic OGC API Asynchronous behavior
- We need a generic Subscription Management API
- 1) Client to create a subscription
- 2) User Prefer header linking subscription
- 3) Server tells what is followed (sync or async with subscription, or nothing at all)
- Pattern can be applied to existing infrastructures as Subscription Management API is an extra ”add-on”

Thank You!

It is important,
to do security right…

Secure Dimensions GmbH
Holistic Geosecurity
Dr. Andreas Matheus

Waxensteinstr. 28
D-81377 München, Germany

Phone	+49 (0)89 38151813-0
Mobile	+49 (0)160 1066366
Telefax	+49 (0)89 38151813-9
Email	am@secure-dimensions.com
Web	www.secure-dimensions.com