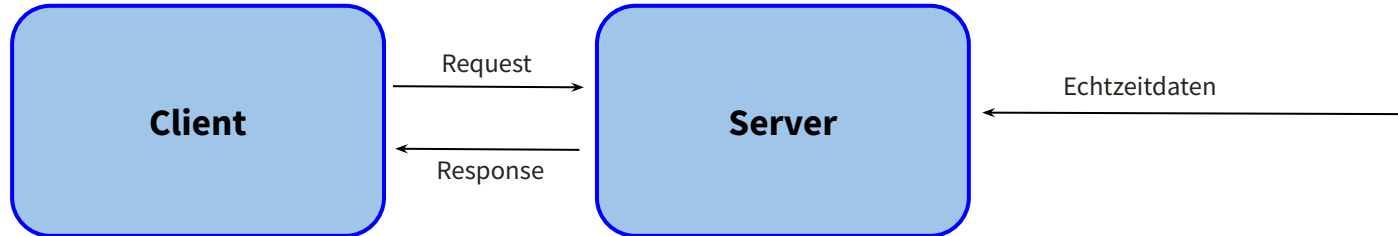


Push-basierte Datenauslieferung mit OGC API Connected Systems und Pub/Sub

Dr. Simon Jirka, 52°North GmbH, jirka@52north.org

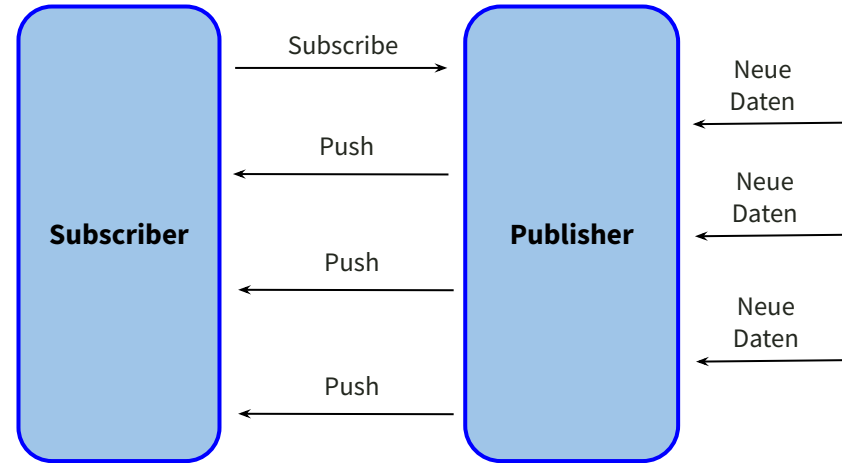
Motivation

- Typisches Zugriffspattern in Geodateninfrastrukturen: Pull/Polling
→ Client fragt regelmäßig beim Server nach“ (Pull/Polling)
- Probleme: Verzögerungen, unnötiger Datenverkehr, hohe Last bei vielen Anfragen
- Gleichzeitig steigender Bedarf an Echtzeitdaten



Publish/Subscribe (Pub/Sub)

- Publish/Subscribe: System verschickt aktiv Nachrichten, wenn etwas Relevantes passiert
- Datenproduzenten („Publisher“) senden Ereignisse an Themenkanäle (Topics)
- Nutzer („Subscriber“) abonnieren relevante Themenkanäle (Topics)
- Aktuelle Informationen werden automatisch ausgeliefert
- Minimierung von Latenzen



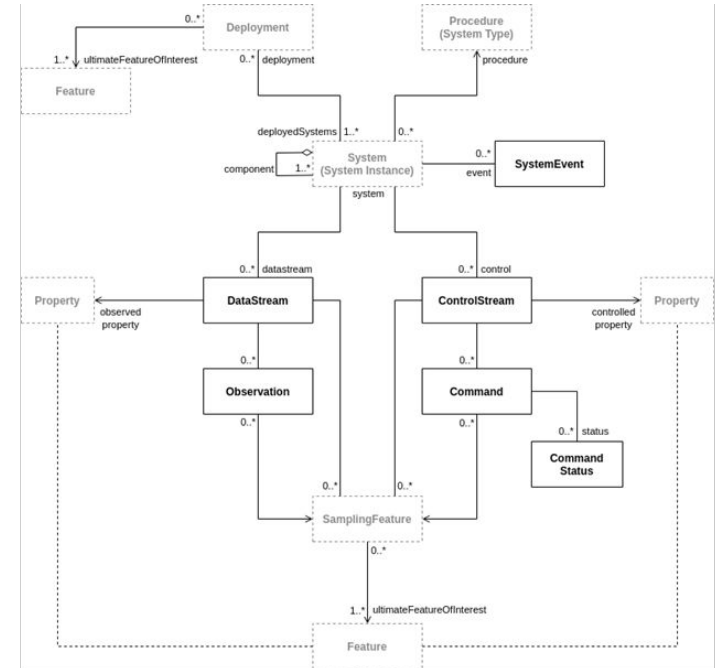
OGC API - Connected Systems

- Spezialisierung der OGC API - Features für
 - Metadaten zu Sensoren, Mess-Stationen, ...
 - Dynamische Sensordaten
- Vorteile
 - Bereitstellung von Messdaten als OGC API-Standard
 - Ermöglicht umfassende Beschreibung von Sensoren
 - Unterstützung vielfältiger Datenformate und Protokolle
 - GeoJSON
 - SensorML (JSON und XML)
 - O&M (JSON und XML)
 - Protobuf
 - WebSocket
 - MQTT
 - ...



OGC API - Connected Systems

- Teil 1 - Feature Resources
 - Sensoren, Plattformen, Beobachter, Messprozesse, Mess-Standorte, ...
 - Veröffentlicht: Version 1.0
- Teil 2 - Dynamic Data
 - Messdaten, Datenströme, Kommandos zur Sensorsteuerung, ...
 - Veröffentlicht: Version 1.0
- Teil 3: Pub/Sub
 - Push-basierte Datenlieferung
 - Work in Progress
- Teil 4: Sampling Feature Types
 - Work in Progress
- Teil 5: Binary Encoding Formats
 - Work in Progress



<https://openeospatial.github.io/ogcapi-connected-systems/api/part2/standard/23-002r0.html>

OGC API - Connected Systems: Part 3

- Push-basierte Auslieferung von (Sensor-)Datenströmen
- Enge Zusammenarbeit mit der OGC Pub/Sub Standards Working Group
 - Spezifikation von Pub/Sub-Mechanismen konsistent über alle OGC API-Standards hinweg (nicht nur Sensordaten)
 - Baut auf der AsyncAPI-Spezifikation auf
 - Beschreibung von Pub/Sub-APIs in einem maschinenlesbaren, interoperablen Format, z.B. Channels/Topics, Protokolle, Nachrichten-Typen
 - Äquivalent zu OpenAPI für asynchrone Kommunikation



OGC API - Connected Systems: Part 3

- OGC API ist unabhängig von spezifischen Protokollen
- Nutzbare Protokolle, z.B.
 - MQTT
 - Wichtiger Standard, z.B. im Internet of Things-Umfeld
 - WebSockets
 - Echtzeit-Webanwendungen.
 - AMQP
 - Beispielsweise für Enterprise-Infrastrukturen

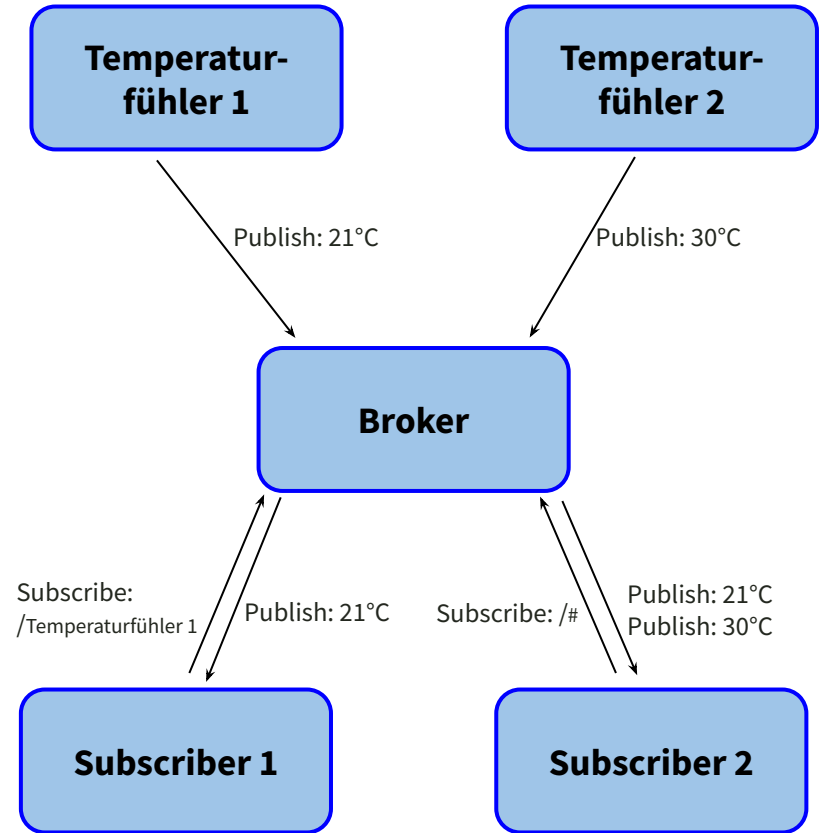


OGC API - Connected Systems: Part 3

- Bezug zu Teil 1 und 2
 - Part 1: Metadaten zu Sensoren/Standorten und Messdaten
 - Part 2: Pull-basierter Zugriff (auf historische Messdaten)
 - Part 3: Push-basierte Auslieferung

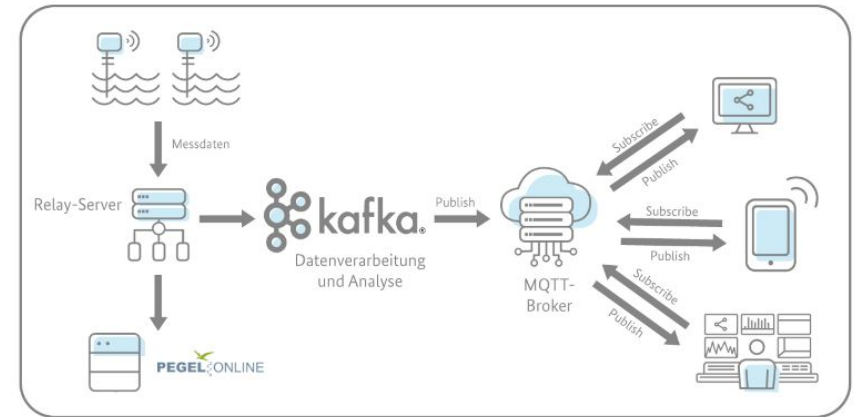
Beispiel: MQTT

- MQTT: Message Queuing Telemetry Transport
- Offenes Protokoll für den Nachrichtenaustausch nach dem Publish/Subscribe Prinzip
- Sender veröffentlicht (Publish) Nachricht bei Broker
- Empfänger abonniert (Subscribe) Nachrichten
- „Abonnements“ von Topics
- Sender und Empfänger sind entkoppelt
- Einfaches Protokoll und kleiner Header



Anwendungsbeispiel

- EDIS: Echtzeitdateninfrastruktur
- Kooperation von ITZBund und 52° North
- Push-basierte Auslieferung von Messdaten aus PEGELONLINE
- Verwendung von MQTT
- <https://www.itzbund.de/DE/itloesungen/egovernment/echtzeitdateninfrastruktur/edis.html>



Quelle: ITZBund

Zusammenfassung

- OGC API - Connected Systems, Part 3 erlaubt Push-basierte Auslieferung von Messdaten
- Relevant für Nah-Echtzeitanwendungen
 - Minimierung von Latenzen und Server-Last
- Flexible Auswahl von Protokollen
 - MQTT, WebSocket, AMQP
 - Je nach Bedürfnissen der konkreten Anwendungsfälle
 - Zuverlässigkeit, Einsatzbereich, Overheads, ...

Vielen Dank für die Aufmerksamkeit

jirka@52north.org