



QGIS Web Client 2 (QWC2) Neues aus dem Projekt

Dr. Horst Düster
@moazagotl
Sourcepole AG, Zürich
www.sourcepole.ch



SOURCEPOLE
Linux & Open Source Solutions



OGIS Web Client 2

- **Ziele OGIS Web Client (OWC2):**
 - Modernes, responsive UI
 - Fokus auf Usability
 - Skalierbare Architektur
 - Modularer Code
 - State of the art Technologie: ReactJS, WebPack, OpenLayers 6





OGIS Web Client 2

kanton glarus geodatenviewer

Adressen, Flurnamen, Grundstücke, Koordinaten...

Adressen, Flurnamen, Grundstücke, Koordinaten...

Adressen, Koordinaten...

Kartenthema auswählen

Kartenebenen zusammenstellen

Karten-Link erstellen und teilen

Hilfe, Dokumentation, Impressum

Kartenthema auswählen

Kartenebenen zusammenstellen

Karten-Link erstellen und teilen

PDF-Karte oder Rasterbild drucken

Kartenwerkzeuge

Hilfe, Dokumentation, Impressum

Karten & Werkzeuge

Kartenthema auswählen

Kartenebenen zusammenstellen

Karten-Link erstellen und teilen

PDF-Karte oder Rasterbild drucken

Kartenwerkzeuge

Hilfe, Dokumentation, Impressum

Werkzeuge
Die einzelnen Werkzeuge werden wie bisher als frei schwebende Panels geöffnet oder nach unten aufgeklappt

Hintergrundkarten

- kein Hintergrund
- Luftbild (Orthofoto)
- farbige Karte
- Graustufenkarte

2 km

Koordinaten [m]: 2713353 / 1210363

Geoportal Kanton GL Nutzungsbedingungen



QGIS Web Client QWC2 (minimal)



QGIS Web Client



- QGIS Project**
- Layers
 - Styles
 - Print templates



QGIS Server

- **Publikation von QGIS Desktop Projekten als WMS, WFS und WCS**
- **Erste Entwicklung 2006 am Institut für Kartographie, ETH Zürich**
- **Integriert in QGIS Codebasis seit 2010**
- **C++, FCGI Applikation**
- **Backend für QGIS Web Client und QGIS Cloud**



QGIS Server Feature

- › Einfache Konfiguration von Projekten in QGIS Desktop
- › Kartensymbologie und Beschriftung identisch zu QGIS Desktop (gemeinsame Rendering Engine)
- › Service für zusätzliche Konfigurationinformationen (GetProjectSettings)
- › PDF-Druck basierend auf QGIS Projekten
- › Datenfilter und Feature-Selektion
- › GetLegendGraphics Erweiterungen



› Kartenkomponenten

- › ReactJS + Redux
- › OpenLayers 6

› QGIS Server

› Optional: Serverseite Microservices für Suche, Permalink und viele mehr



Modulare Kernfunktionen

- › Themenwähler (Projekte)
- › Layerbaum
- › Feature info
- › Suche mit konfigurierbaren Providern
- › Messwerkzeuge
- › Redlining Funktionalität
- › Permalink Generierung
- › PDF-Druck
- › Screenshot / Raster Export
- › WMS / WFS Import
- › KML Import
- › Kartenvergleichs-Tool



Erweiterte Funktionen

- › **Funktionen mit Server-Komponenten**
 - › DB-Suche
 - › Permalinks
 - › Reporting
 - › Editierfunktionen
 - › Mapinfo-Service (layerunabhängig)
 - › Legendendruck
 - › Höhenprofile
 - › Selbstregistrierung (Gruppen)
- › **Übersetzt in 7 Sprachen**



Demo der Neuigkeiten



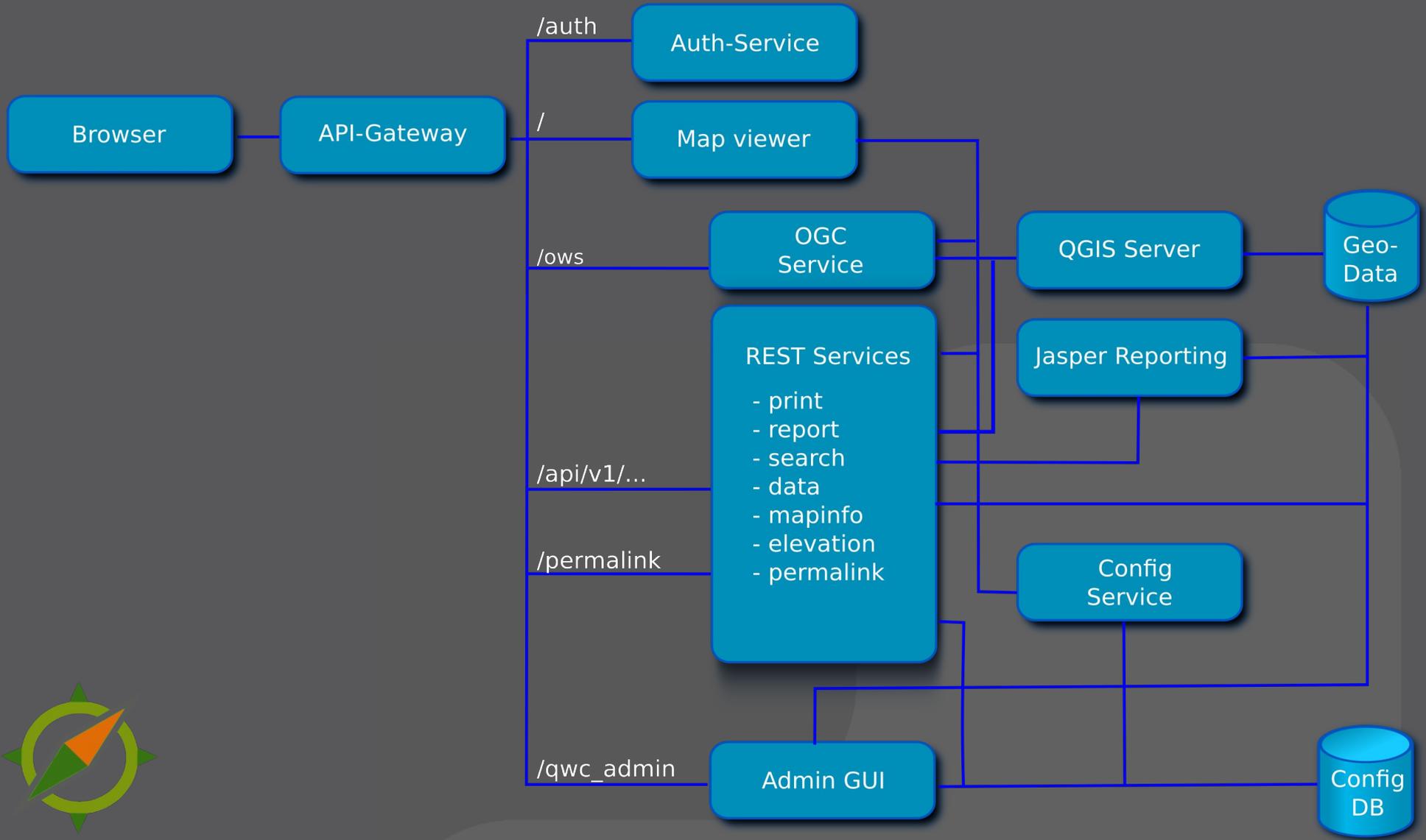
- › **Optionales Backend für QWC2**
- › **Modular, Micro-service orientiert**
- › **(Hauptsächlich) Python Flask Web Applikationen**
- › **Integrierte API Dokumentation (OpenAPI/Swagger)**
- › **Deployment als Docker Container oder WSGI**



The screenshot shows the GitHub profile page for the organization 'qwc-services'. At the top, there is a navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The profile header includes the organization's logo (a green compass) and the name 'qwc-services'. Below this, there are tabs for 'Repositories 15', 'People 3', 'Teams 1', 'Projects 0', and 'Settings'. The main content area is titled 'Pinned repositories' and features two pinned repositories: 'qwc-services-core' (Python, 1 fork) and 'qwc-docker' (Shell, 1 star, 1 fork). Below the pinned repositories is a search bar and filters for 'Type: All' and 'Language: All', along with a 'New' button. The repository list shows 'qwc-config-service' (Python, updated 7 days ago) and 'qwc-map-viewer' (Python, 1 fork, updated 8 days ago). On the right side, there are two sidebars: 'Top languages' showing Python, Shell, and Java, and 'People' showing 3 members, with 'manisandro' (Sandro Mani) listed.



Server Architektur





Kartenviewer und Drucken

- › **QWC map viewer**
 - › QWC2 viewer service mit Zugriffskontrolle
- › **QWC OGC service**
 - › QGIS Server basierter WMS/WFS mit Zugriffskontrolle
- › **QWC Print service**
 - › QGIS Server basierter PDF Druck mit Zugriffskontrolle



Weitere Dienste

- › **QWC Search Service**
 - › Such-API für benutzerdefinierte Suche
- › **QWC Data Service**
 - › Lesen und Editieren von Daten
- › **QWC Mapinfo service**
 - › Layerunabhängige Info (Rechtsklick)
- › **Permalink service**
- › **Elevation service**



QWC Admin GUI

- › **Web-GUI für Konfigurations-DB**
- › **User und Gruppenmanagement**
- › **Rollenbasierter Zugriff auf Ressourcen**
- › **Ressource-Typen:**
 - › Karten (QGIS-Projekte)
 - › Drucktemplates
 - › Layer, Attribute
 - › Daten (create, read, update, delete)
 - › Viewer, Viewer tasks
 - › Benutzerdefinierte Ressourcen



Authentisierung

- › **Modulare Authentisierungs-Dienste**
 - › Stellen JWT Tokens für zugriffsbeschränkte Zugriffe aus (Cookie oder JWT token header)
- › **QWC DB Auth**
 - › Authentisierung mit User-DB
- › **QWC LDAP Auth**
 - › Authentisierung mit LDAP/Active Directory
- › **und weitere (z.B. Kerberos, SAML)**

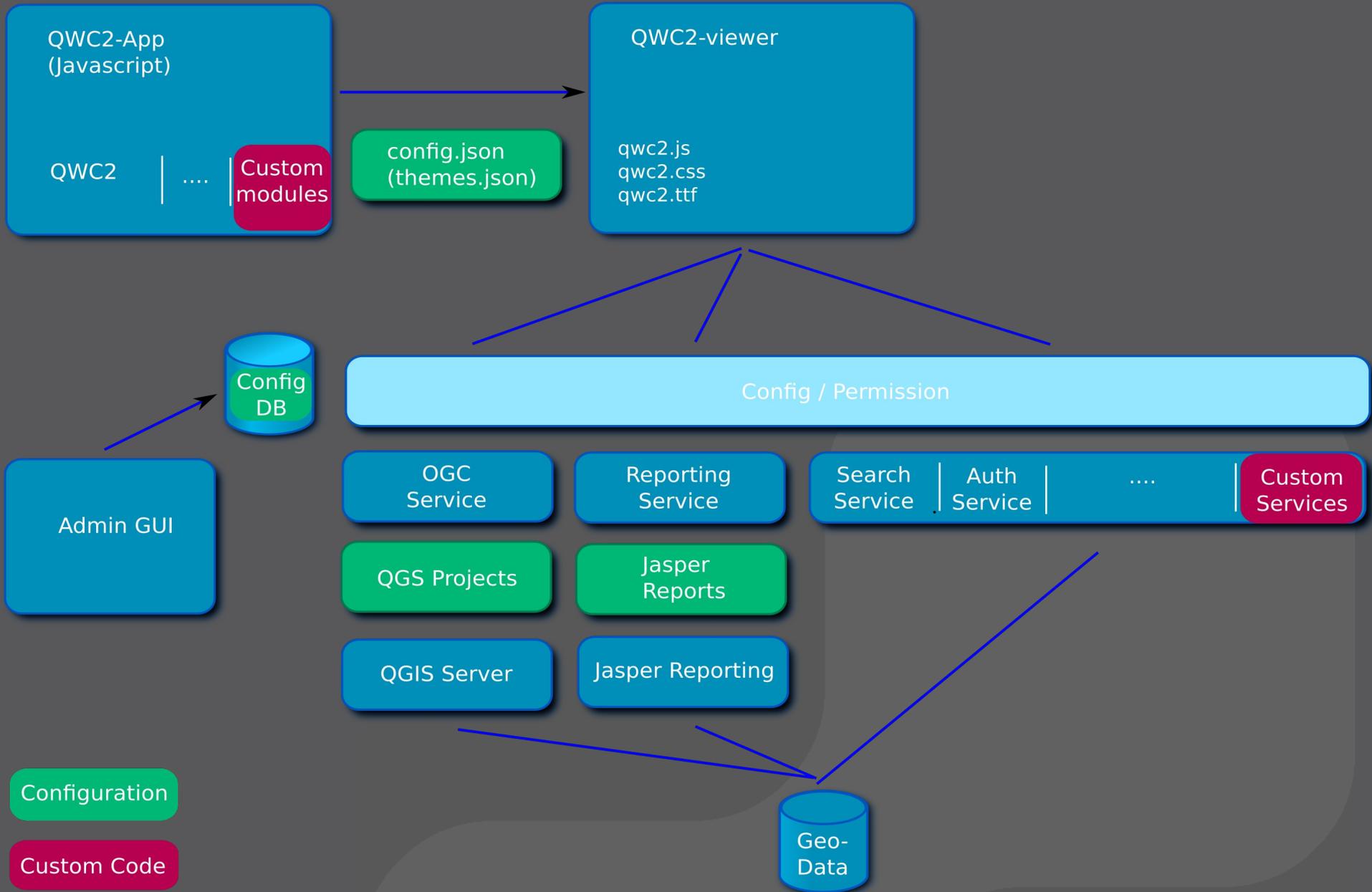


Interne Dienste

- › **QWC Services Core**
 - › Basismodul für QWC services und Dokumentation
- › **QWC Config Service**
 - › Zugriffsrechte und User-spezifische Konfiguration
- › **QGIS Server**
- › **Jasper Reporting Service**
 - › Jasper Reports web service



Customizing





Micro Services

- **Komplexe Anwendung aus lose gekoppelten Teilprozessen (Services)**
- **Isolierte, überschaubare Funktionalität in Services**
- **Kommunikation zwischen Services über HTTP/REST**
- **<https://de.wikipedia.org/wiki/Microservices>**



Docker / Kubernetes

› Docker

- › Bereitstellung von Anwendungen mit Containervirtualisierung
- › 1 Service pro Container
- › Einfache Skriptsprache zur Erstellung
- › Verteilung von Images über Registry
- › <https://www.docker.com/>

› Kubernetes

- › System zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von Container-Anwendungen
- › <https://kubernetes.io/>



Vor-/Nachteile Micro-Services

- › **Aufwändiger Betrieb (Anzahl Services, DB-Connections, Logs, ...)**
- › **Feingranulare Skalierbarkeit**
- › **Separierung Authentisierung+ Autorisierung**
- › **Modulare Authentisierung**
- › **Services mit verschiedenen Technologien**
- › **Modulare Migration von Services**
- › ...



› QWC2 Dokumentation

- › <https://qwc2.sourcepole.com>

› Beispiele:

- › <https://qgiscloud.com>
- › <https://map.geo.gl.ch>
- › <https://geo.so.ch/map/>



Danke!



Dr. Horst Düster
@moazagotl