QWC2 viewer
For QGIS server
with micro service architecture

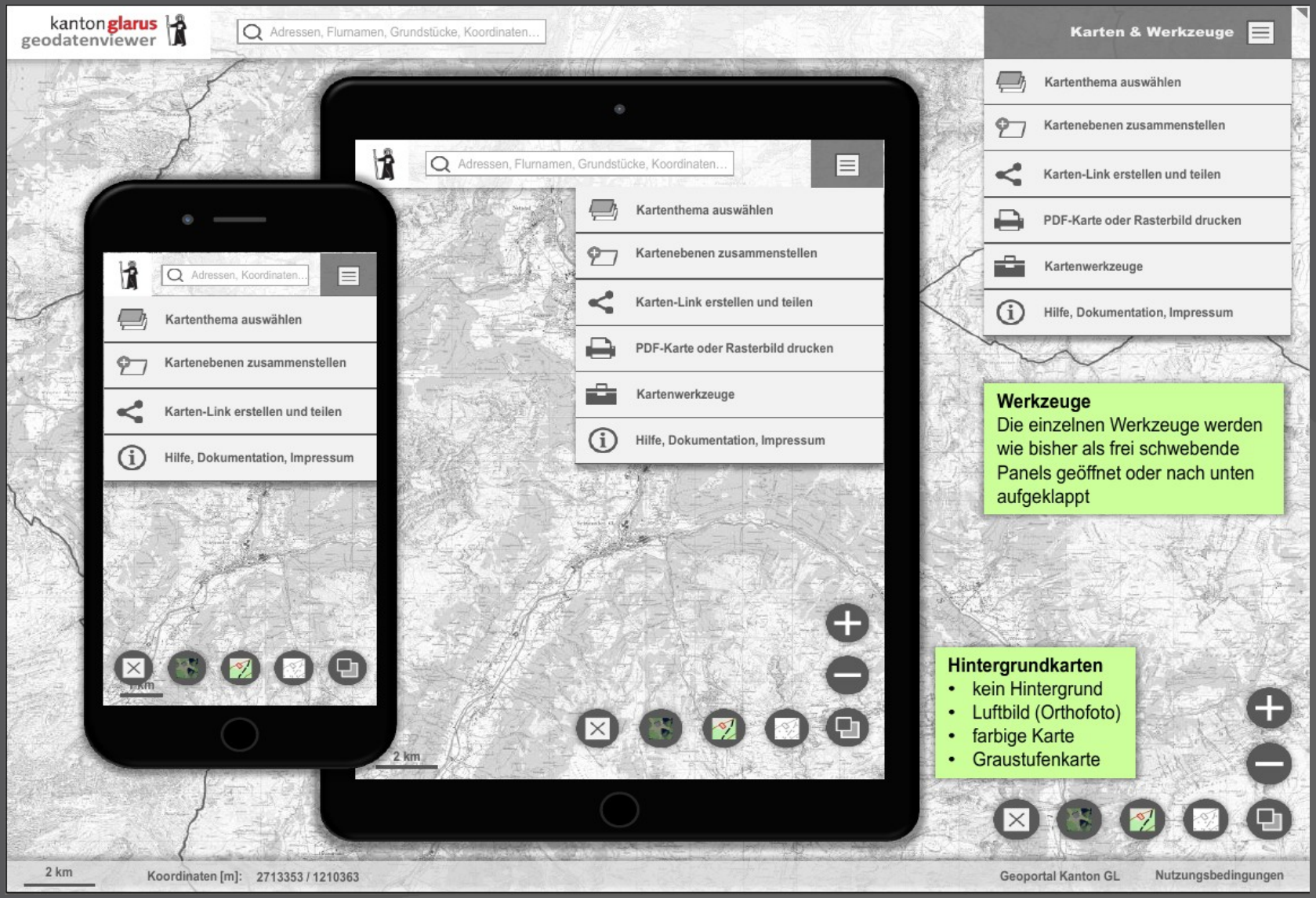
Pirmin Kalberer
@implgeo
Sourcepole, Switzerland
www.sourcepole.com

QGIS Web Client 2

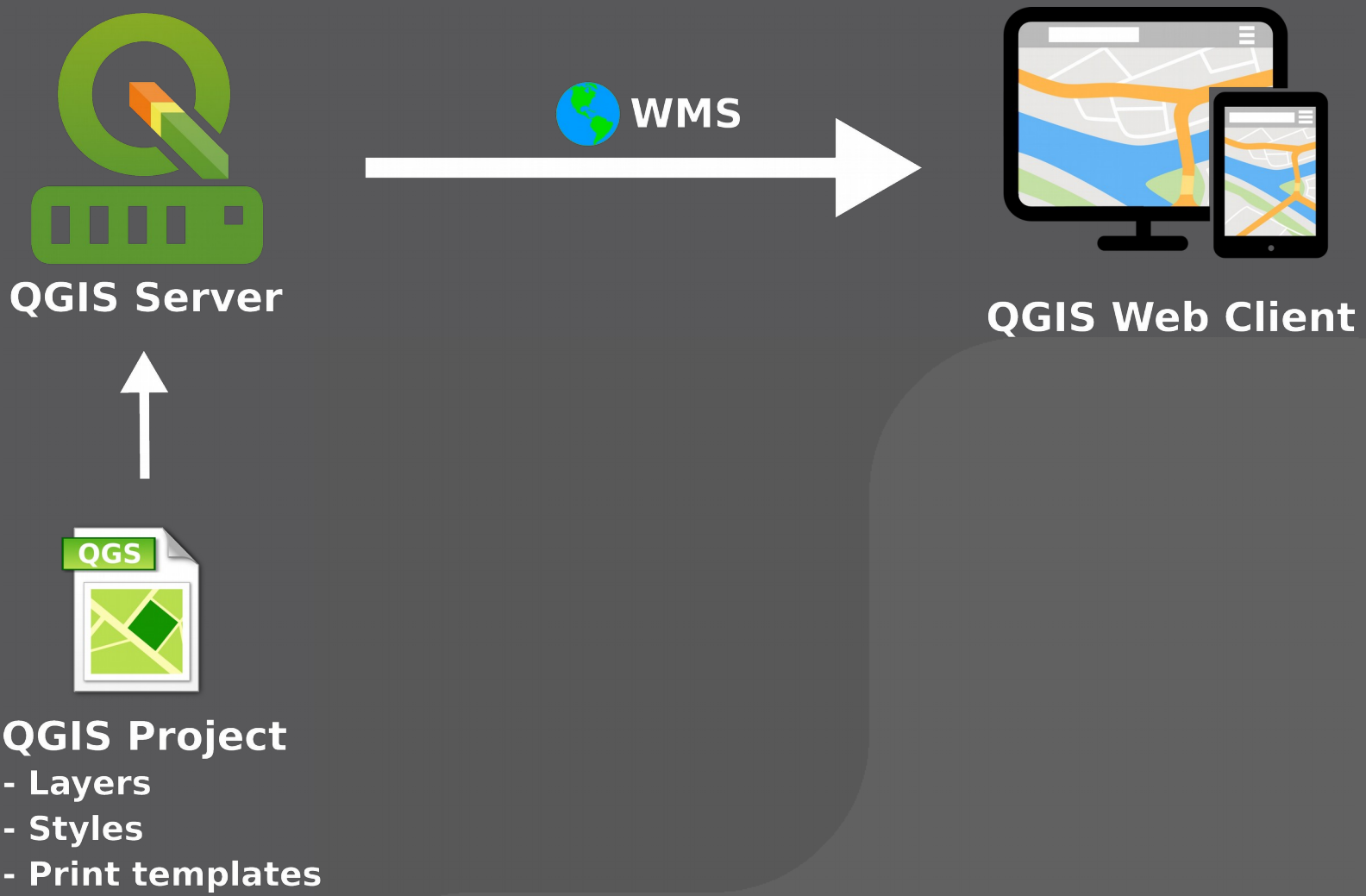
- Goals
  - Modern, responsive interface
  - Less complexity → Focus on usability
  - Scalable architecture
  - Modular code base
  - State-of-the art technology: ReactJS, OpenLayers 5

QGIS Web Client 2

QGIS Web Client 2



QGIS Web Client OWC2 (minimal)



QWC2 technology

- **Map components**
  - ReactJS + Redux
  - OpenLayers 5
- **Build tool chain: nodejs / yarn / webpack**
  - QWC2 demo application
- **QGIS Server**
- **Optional: server side services for search, permalink, etc.**

Core modules

- Theme (project) browser
- Layer tree
- Feature info
- Search with configurable providers
- Measure tools
- Sketching / redlining functionality
- Permalink generation
- PDF-Print
- Screenshot / Raster export
- WMS / WFS import
- KML import
- Map comparison tool

Additional features

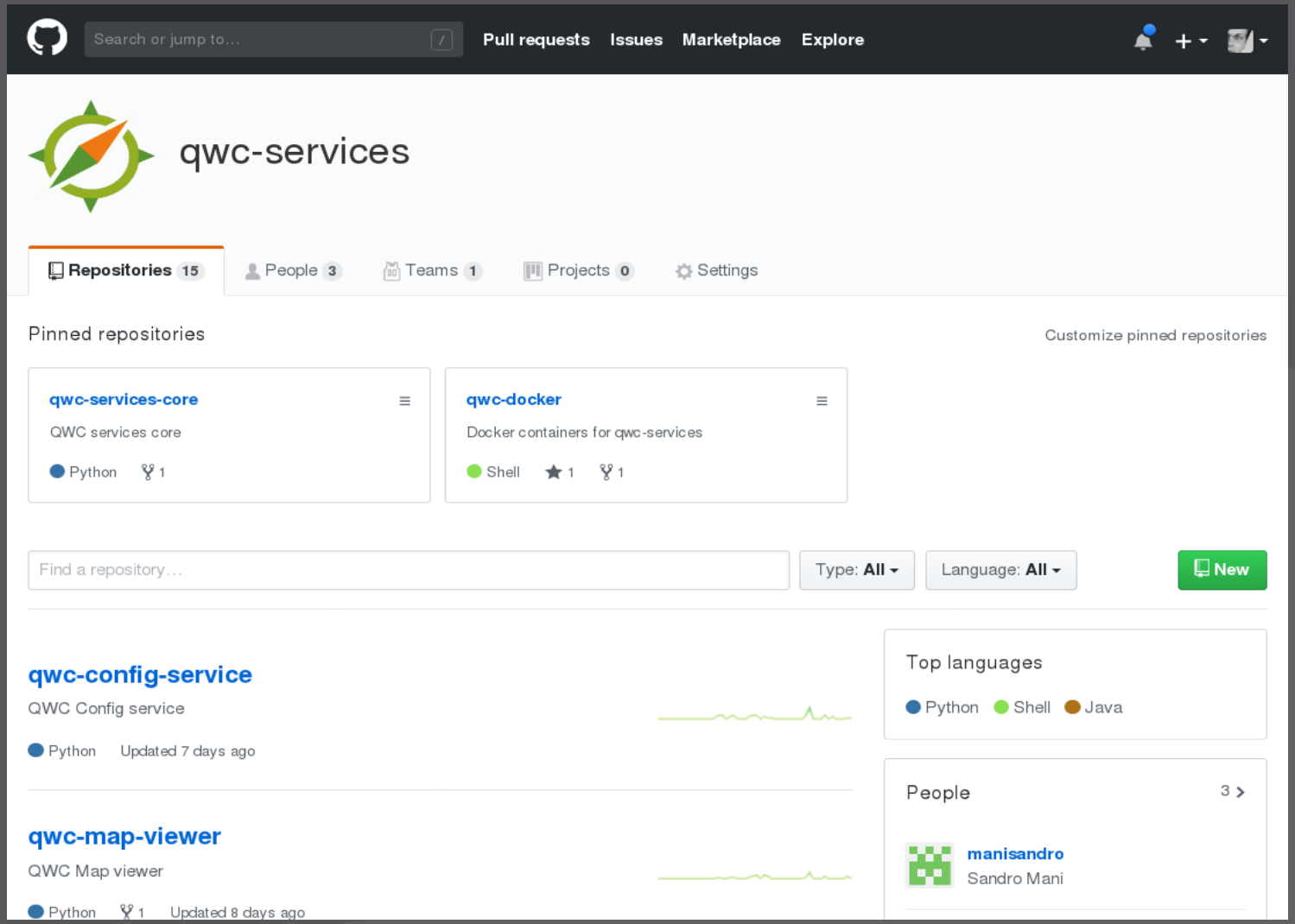
- **Features with server components**
  - DB search
  - Permalinks
  - Reporting
  - Editing
  - Mapinfo service (layer independent)
  - Legend priting
  - Height profiles
  - Self registration (groups)
- **Translated into 7 languages**

Demo

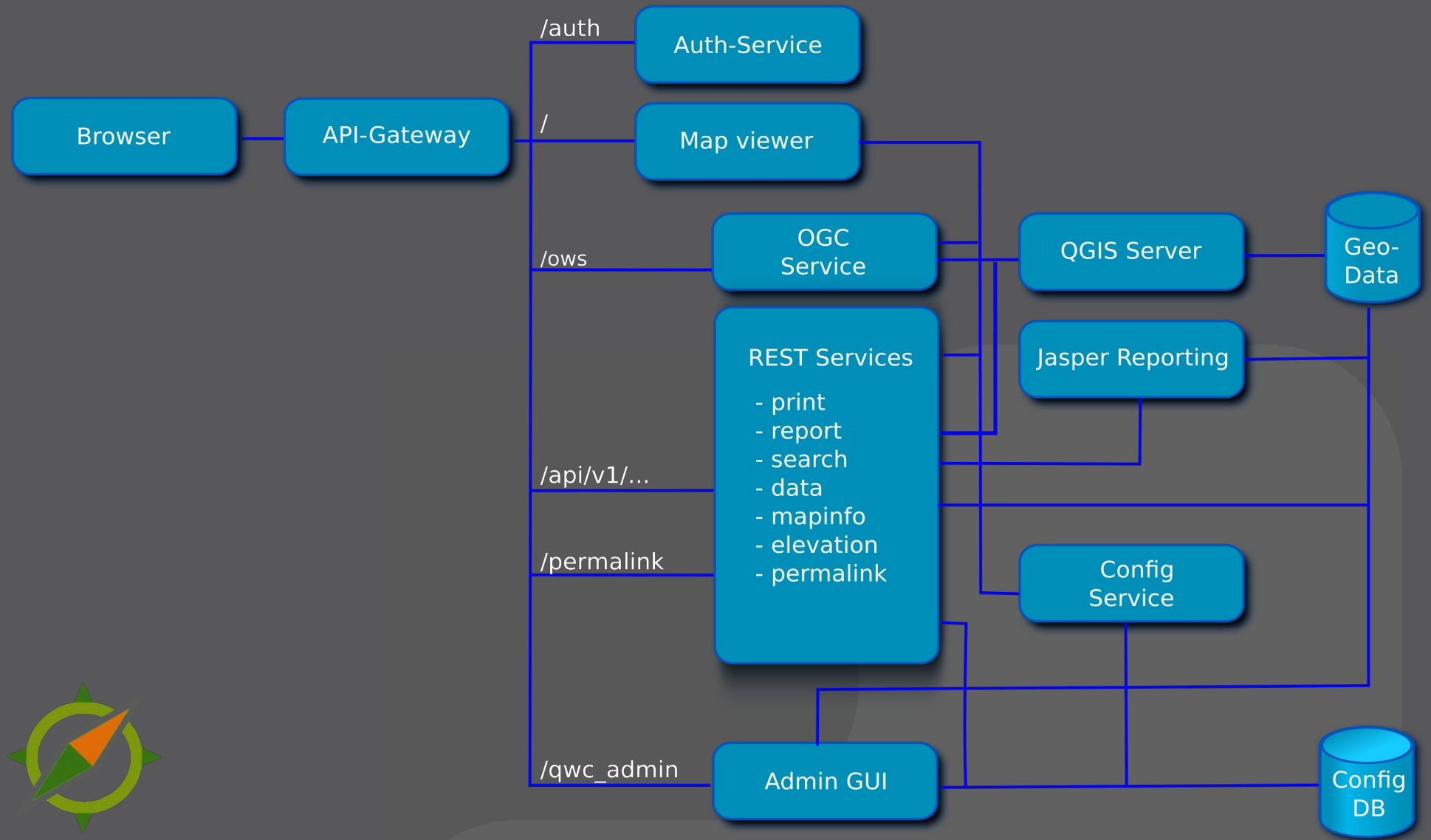
qwc-services

- Optional backend for QWC2
- Modular, micro-service oriented
- (Mostly) Python Flask web applications
- Integrated API documentation (OpenAPI/Swagger)
- Deploy as docker containers or WSGI

github.com/qwc-services



Server architecture



Maps and printing

- **QWC map viewer**
  - QWC2 viewer service with access control
- **QWC OGC service**
  - QGIS Server based WMS/WFS with access control
- **QWC Print service**
  - QGIS Server based PDF printing with access control

Additional services

- **QWC Search Service**
  - Search API for custom DB searches
- **QWC Solr Search Service**
  - Fulltext search with Apache Solr
- **QWC Data Service**
  - Access and edit spatial data
- **QWC Mapinfo service**
  - Layer independent info (right-click)
- **Permalink service**
- **Elevation service**

QWC Admin GUI

- Web GUI for Config DB
- User and group management
- Role based resource access permissions
- Resource types
  - Maps (QGIS projects)
  - Print templates
  - Layer, Attribute
  - Data (create, read, update, delete)
  - Viewer, Viewer tasks
  - Custom resources

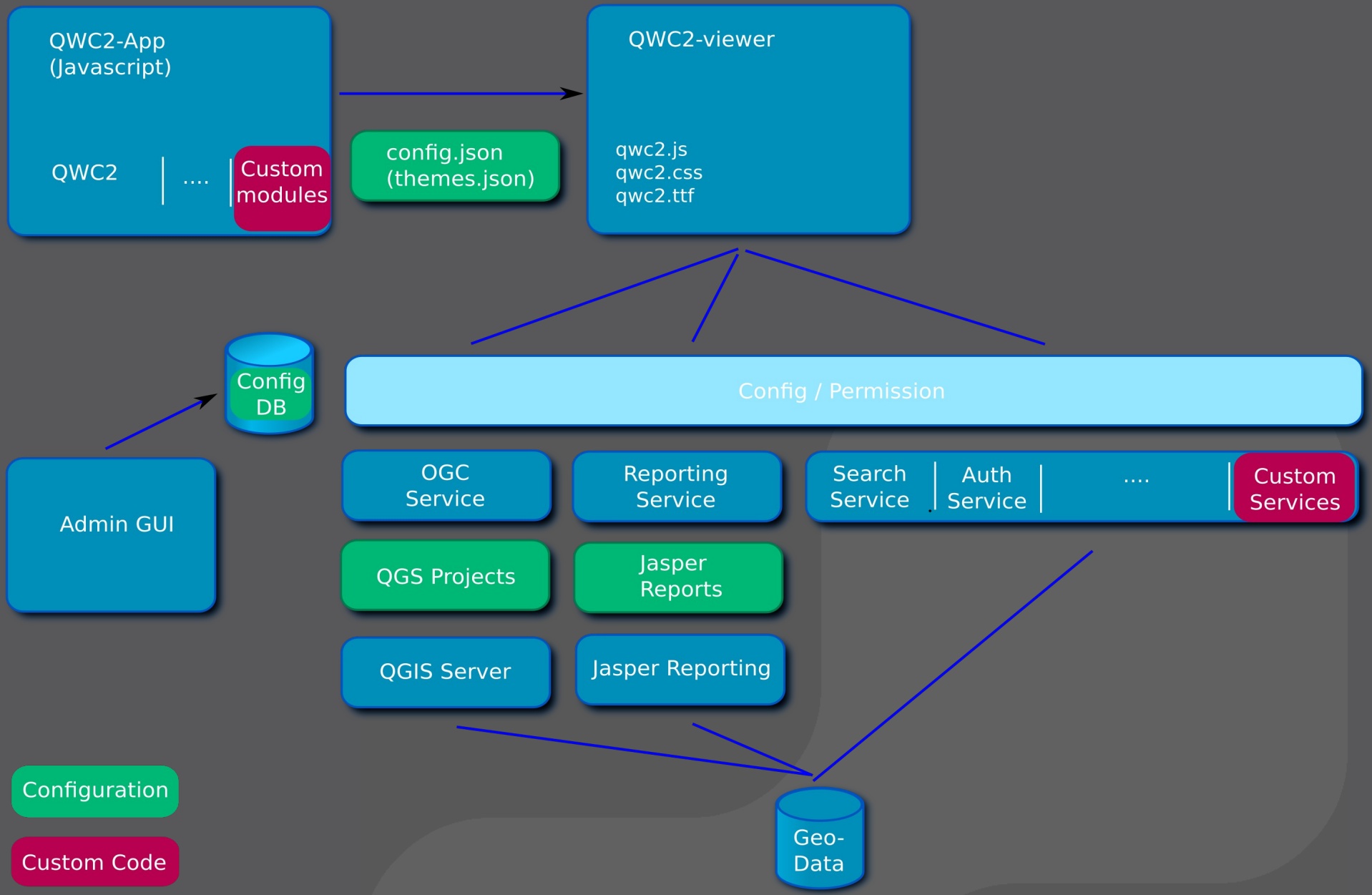
Authentication services

- **Pluggable authentication services**
  - Authentication services issue JWT tokens for privileged map and service access (Cookie or raw JWT token)
- **QWC DB Auth**
  - Authenticate with user database
- **QWC LDAP Auth**
  - Authenticate with LDAP/Active Directory
- **more available (e.g. Kerberos, SAML)**

Internal services

- **QWC Services Core**
  - Shared modules for QWC services and documentation for setup
- **QWC Config Service**
  - Service permissions and user specific configuration
- **QGIS Server**
- **Jasper Reporting Service**
  - Jasper Reports web service

Customizing



Micro services

- Complex applications built from loosely coupled processes (services)
- Isolated, modular functionality in services
- Communication between services over HTTP/REST

OpenAPI/Swagger

Data service API 1.0

[Base URL: /api/v1/data]
https://geo.so.ch/api/v1/data/swagger.json

API for SO!MAP Data service.

General Information for all operations:

Datatypes-Encoding:

JSON only defines recommendations or has no information concerning the encoding of some quite common used database data types. Following a description on how these are encoded in the data service API.

- Date: ISO date strings `YYYY-MM-DD`
- Datetime: ISO date/time strings `YYYY-MM-DDThh:mm:ss`
- UUID: Hex-encoded string format. Example: `'6fa459ea-ee8a-3ca4-894e-db77e160355e'`

Feature-ID:
For operations like updating or deleting features, records are identified by a feature `id`. This `id` refers to the primary key of the database table and is usually kept constant over time.

default Data edit operations

POST	/{dataset}/	Create a new dataset feature
GET	/{dataset}/	Get dataset features
PUT	/{dataset}/{id}	Update a dataset feature
DELETE	/{dataset}/{id}	Delete a dataset feature
GET	/{dataset}/{id}	Get a dataset feature

Docker / Kubernetes

- **Docker**
  - Packaging of applications with container virtualization
  - Single service per container
  - Simple scripting language for building
  - Distribution of images via container registry
  - https://www.docker.com/
- **Kubernetes**
  - System for automatization of controlling, scaling and maintenance of containerized applications
  - https://kubernetes.io/

Pro-/contra Micro-Services

- 👎 High effort for operations (Number of services, DB connections, logging, …)
- 👍 Fine granular scaling
- 👎 Separation of authentication and authorization
- 👍 Modular authentication
- 👍 Services with mixed technologies
- 👍 Modulare migration of services
- …

QGIS Web Client 2

- **Source code and issue tracker**
  - https://github.com/qgis/qwc2-demo-app
  - https://github.com/qgis/qwc2
  - https://github.com/qwc-services/qwc-services-core
  - https://github.com/qwc-services/qwc-docker
- **QWC2 documentation**
  - https://github.com/qgis/qwc2-demo-app/blob/master/doc/QWC2_Documentation.md
- **Examples**
  - https://map.geo.gl.ch
  - https://geo.so.ch/map/
  - https://qgiscloud.com

Thank you!

Pirmin Kalberer
@implgeo