



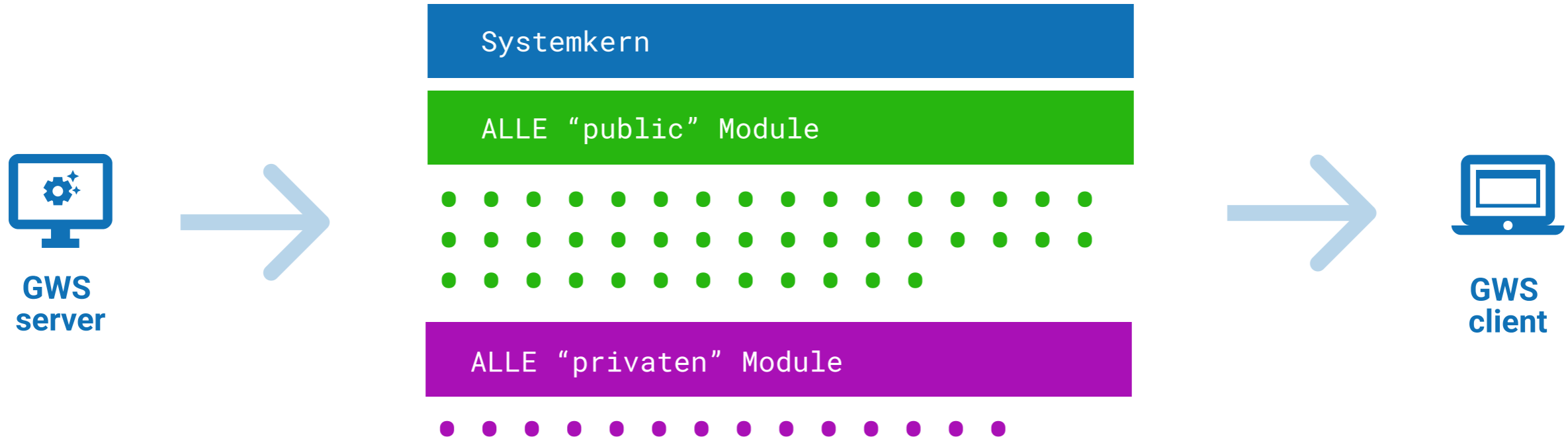
GBD WebSuite Plugins: Administration und Entwicklung

Georg Barikin
Geoinformatikbüro Dassau GmbH

barikin@gbd-consult.de

GBD WebSuite r7

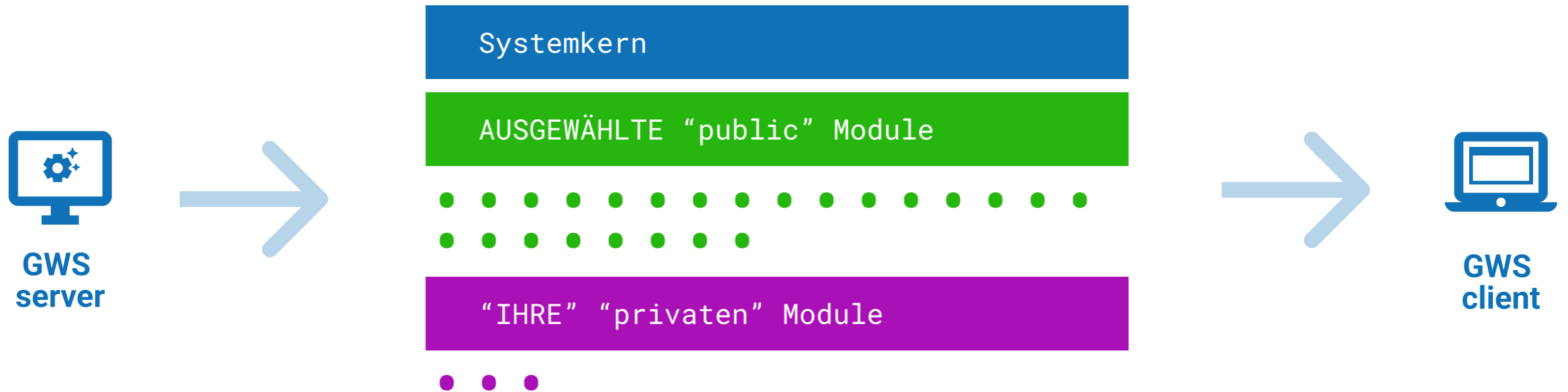
Modularer Aufbau, aber alles in einem "Topf"



es werden immer alle Module geladen

GBD WebSuite r8

Ein Plugin-System



Ihre Installation hat nur die Module, die Sie tatsächlich nutzen

Praktisches Beispiel

Im Gelände sind Sensoren platziert (z.B. Wetter)

Es gibt eine REST API zum Abruf von Daten

Daten als Punktenlayer darstellen

und per Mausklick aktualisieren

Technische Umsetzung

- einen Namen wählen
- Ordner erstellen
- Ordner muss von Docker erreichbar sein

```
/var/gws
```

```
  /gws-server
```

```
    /app
```

```
    /doc
```

```
    README.md
```

```
  /data
```

```
    /config
```

```
      config.cx
```

```
    /projekte
```

```
    /www
```

```
    /plugins
```

```
      /sensors
```

Plugin Ordner

- Server Umsetzung: python
- Client Umsetzung: *tsx und *css
- Tests
- Dokumentation

```
/plugins/sensors
```

```
  __init__.py
```

```
  /js
```

```
    index.tsx
```

```
    index.css
```

```
  /test
```

```
    api_test.py
```

```
    network_test.py
```

```
    utils_test.py
```

```
  /doc
```

```
    user-de.doc.md
```

```
    admin-de.doc.md
```

Server Umsetzung (python)

- “Config” beschreibt die Konfiguration. Datentypen vom System überprüft!
- Decoratoren sorgen für eine richtige Anbindung
- Command-Methoden sind automatisch im Client verfügbar

```
@gws.ext.config.action('sensors')
class Config:
    apiKey: str
    sensorNames: t.List[str]

@gws.ext.object.action('sensors')
class Object(gws.base.action.Object):
    def configure(self):
        self.apiKey = self.config('apiKey')
        self.sensorNames = self.config('sensorNames')

@gws.ext.command('sensorsGetData')
def get_data(self, request):
    features = get_features_from_api()
    return gws.Response(features=features)

@gws.ext.object.layer('sensors')
class Object(gws.base.layer.vector.Object):
    def get_features(self, bounds):
        return get_features_from_api()
```

Client Umsetzung (typescript)

- Server Aktionen automatisch verfügbar
- Einbindung durch “tags” und Registrierung
- Client API noch nicht stabil!

```
class SensorsToolbarButton extends toolbar.Button {
  iconClass = 'sensorsToolbar';

  async updateSensors() {
    let response =
      await this.app.server.sensorsGetData();

    layer.replaceFeatures(response.features);
  }

  whenTouched() {
    this.updateSensors()
  }
}

gws.registerTags({
  'Toolbar.Sensors': SensorsToolbarButton,
});
```


Konfiguration

- “Aktion” und “Layer” konfigurieren
- Client “Element” hinzufügen

```
project {  
  
  actions+ {  
    type "sensors"  
    apiKey "hello"  
    sensorNames [ "eins" "zwei" "drei" ]  
  }  
  
  map {  
    layers [  
      ...  
      {  
        type "sensors"  
        title "Wetter Sensoren"  
      }  
      ...  
    ]  
  }  
  
  client.elements [  
    ...  
    { tag "Toolbar.Sensors" }  
    ...  
  ]  
}
```

Plugin einbinden (MANIFEST.json)

- Plugins müssen explizit in der Manifest Datei freigeschaltet werden
- bei allen Admin Aufgaben ist die Manifest Datei anzugeben

```
{
  "locales": [
    "de_DE"
  ],
  "release": "8.0",
  "plugins": [
    {
      "name": "sensors",
      "path": "/data/plugins/sensors",
      "hostPath": "/hdd/arbeitsgws/plugins/sensors"
    }
  ],
  "withStrictConfig": true,
  "withFallbackConfig": false
}
```

- Mit “make test” + MANIFEST werden die Tests in der gws Test-Suite mit ausgeführt

```
/hdd/gws$ make test MANIFEST=/hdd/arbeit/gws/MANIFEST.json
```

```
api_test.py::test_request_ok PASSED [ 30%]  
api_test.py::test_request_redirect_ok PASSED [ 40%]  
api_test.py::test_request_404 PASSED [ 50%]
```

Dokumentieren

- Dokumentation im Markdown Format
- Mit "Section IDs" werden die Docs an der richtigen Stelle eingefügt

```
# sensors :/admin-de/config/layer/
```



GBD WebSuite Dokumentation

Administrator Handbuch

Einführung

Konfiguration A-Z

Drucken

Features

Karten

Layer

Lokalisierung

Metadaten

Layer Typen

geojson



sensors



wms



wmts

Teilen

- nach der Open-Source-Philosophie ist es immer vorteilhaft, Ihre Entwicklungen mit der Welt zu teilen
- ist jedoch lizenztechnisch keine Pflicht
- ab r8 werden wir Pull Requests annehmen

