

# Fostering Research

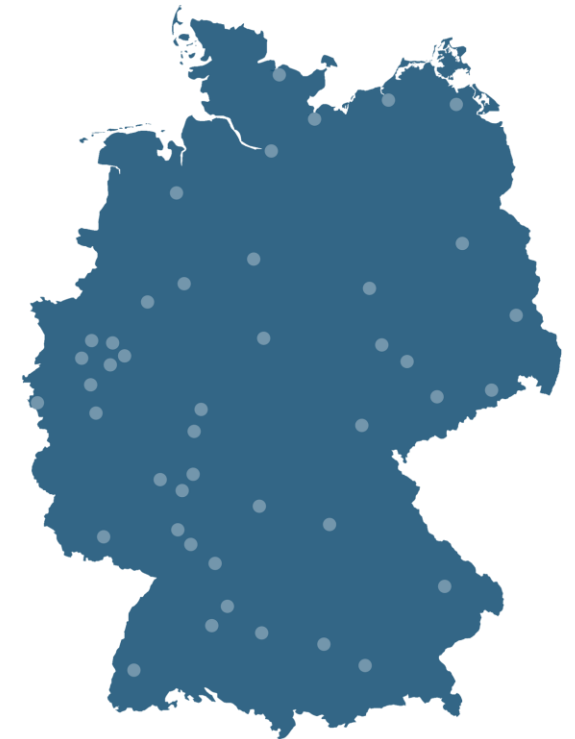
## Real World Data Sharing and Process Orchestration with FHIR and BPMN

HL7 FHIR DevDays - Amsterdam, June 2025

Hauke Hund, GECKO Institute, Heilbronn University of Applied Sciences

# Motivation

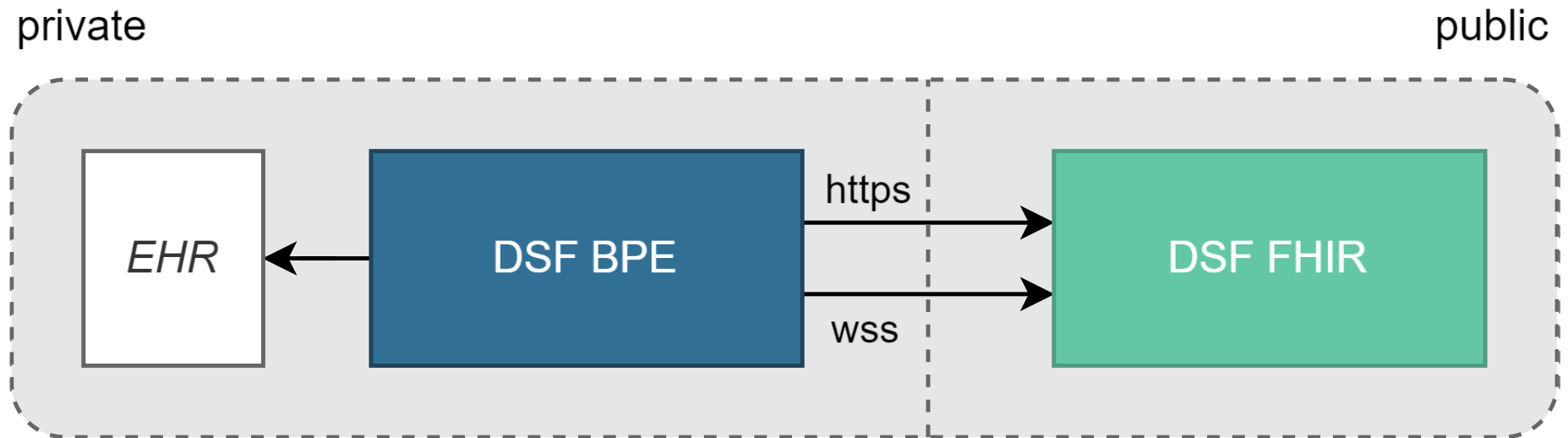
- Enhanced research collaboration
- Standardized data access and usage
- Interoperable solution
- Data discovery with feasibility queries
- Coordination of data requests and approvals
- Data transport, consolidation, pseudonymization and distribution
- Distributed data analysis



# Data Sharing Framework

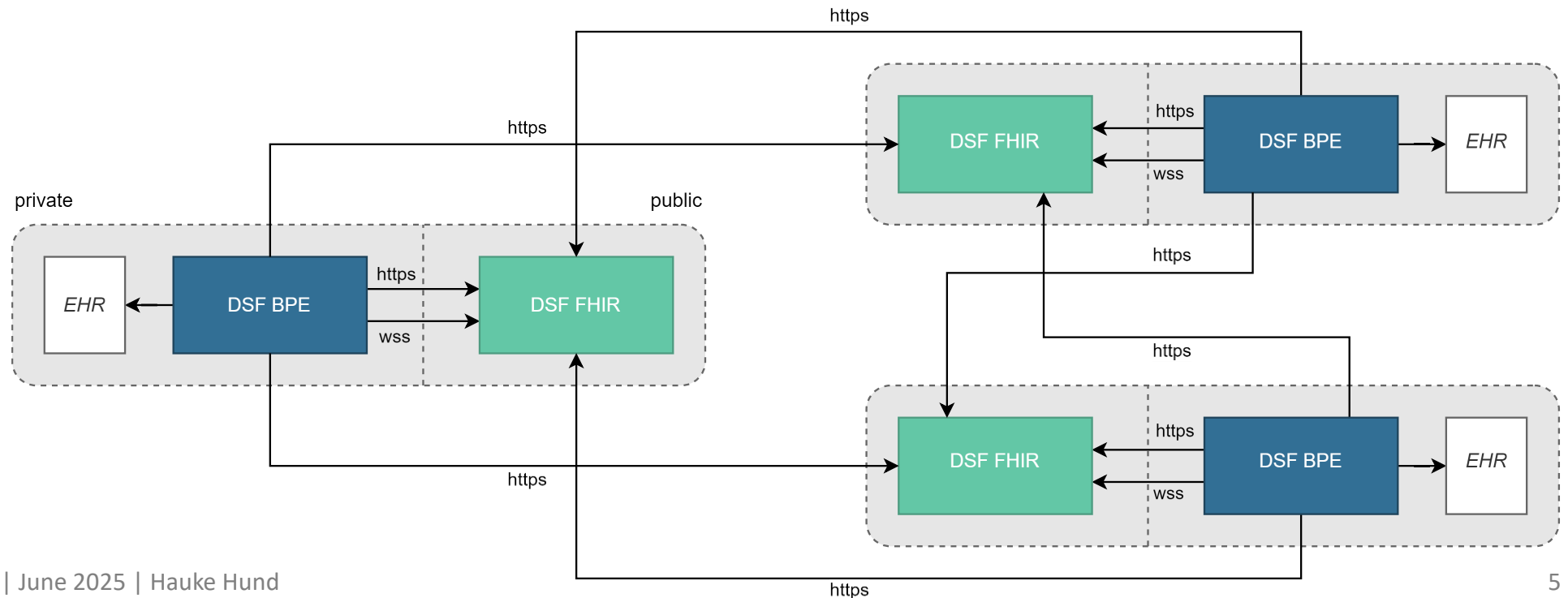
**Public:** FHIR Server “Mail Box”

**Private:** Business Process Engine



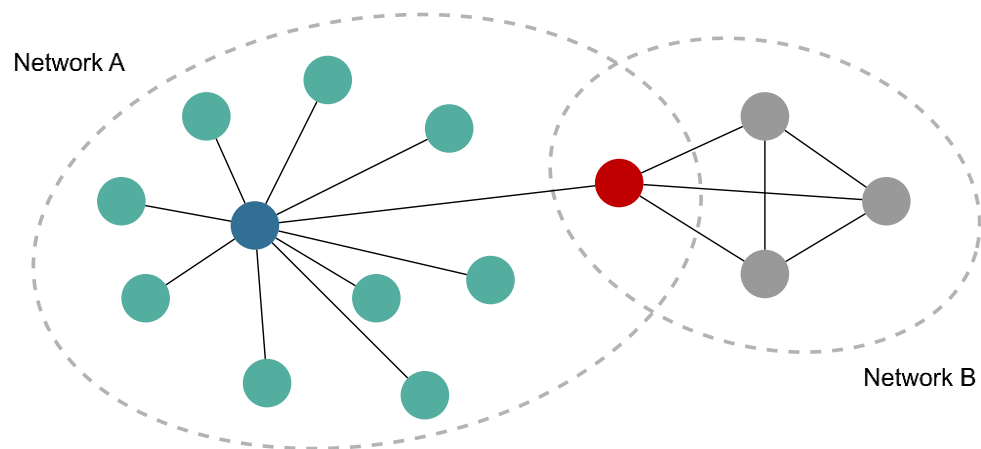
# Data Sharing Framework

*A distributed business process engine to coordinate and exchange medical data in healthcare research and delivery*

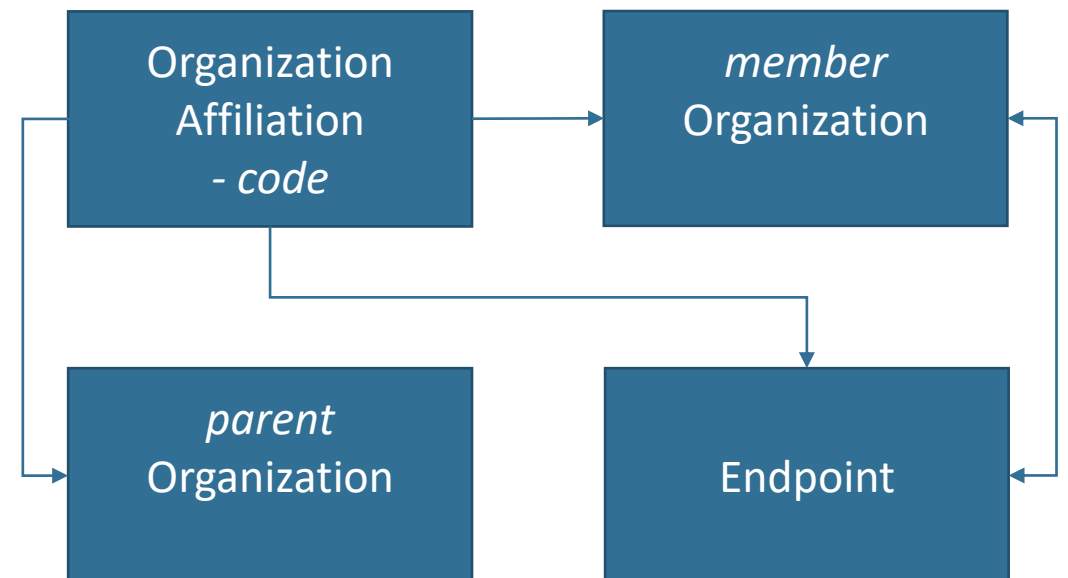


## Research Networks by Configuration

One DSF instance can operate in multiple research / healthcare delivery networks



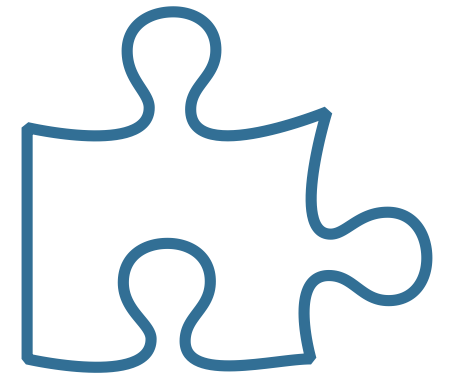
Allow-List configures roles in network



## Process Plugins

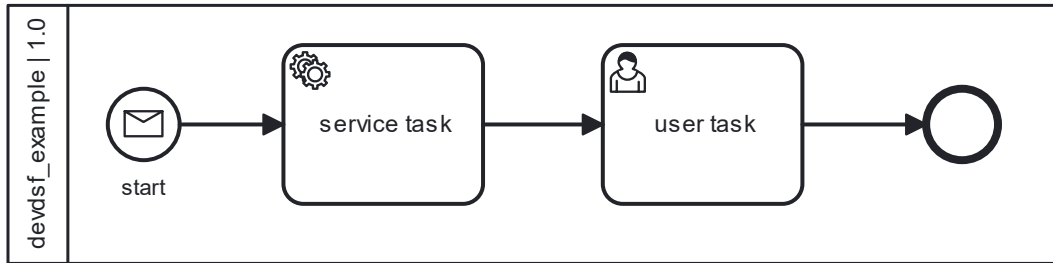
---

- Use case specific data structures and logic
- A process plugins is an archive with
  - BPMN 2.0 models
  - FHIR R4 resources
  - Java Code
- Deployed in the DSF BPE as a Jar file
  - Web Application Style Class Loading
  - Same process, multiple versions

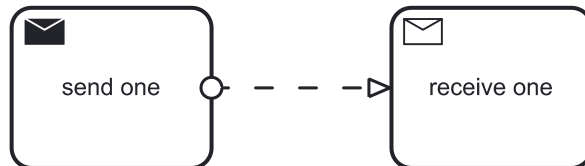


# BPMN and FHIR

## Process



## Message Send / Receive Tasks



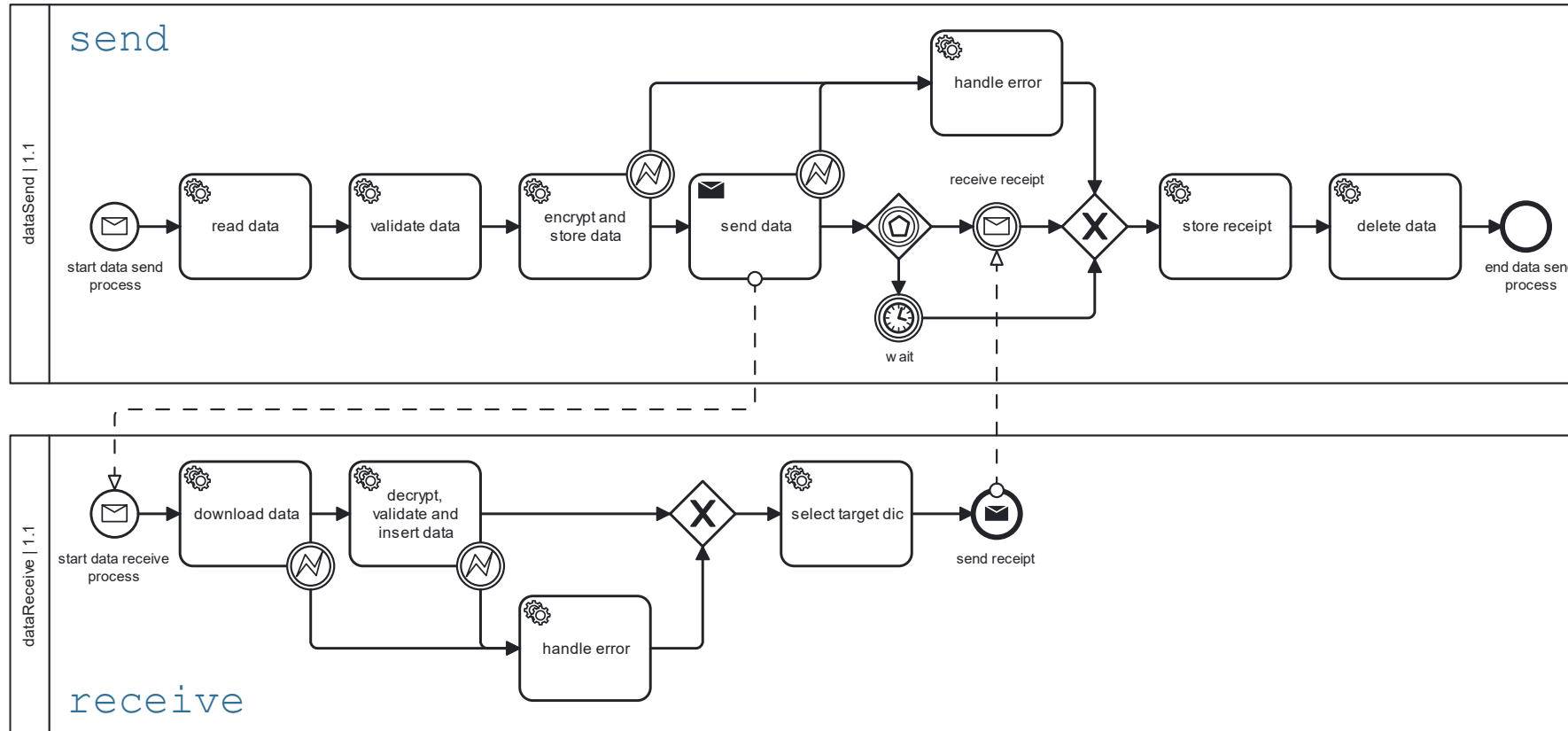
## Message Events



```

<ActivityDefinition xmlns="http://hl7.org/fhir">
  <extension url="http://dsf.dev/fhir/
    StructureDefinition/extension-process-authorization">
    <extension url="message-name">
      <valueString value="start"/>
    </extension>
    <extension url="task-profile">
      <valueCanonical value="http://dsf.dev/fhir/
        StructureDefinition/example|1.0"/>
    </extension>
    <extension url="requester">
      <!-- all, organization, role in parent org -->
    </extension>
    <extension url="recipient">
      <!-- all, organization, role in parent org -->
    </extension>
  </extension>
  <url value="http://dsf.dev/bpe/Process/example"/>
  <version value="1.0"/> <!-- managed by DSF BPE -->
  <status value="active"/> <!-- managed by DSF BPE -->
  <date value="2025-06-06"/> <!-- managed by DSF BPE -->
  <kind value="Task"/>
</ActivityDefinition>
  
```

# BPMN: Encrypted Data Transfer

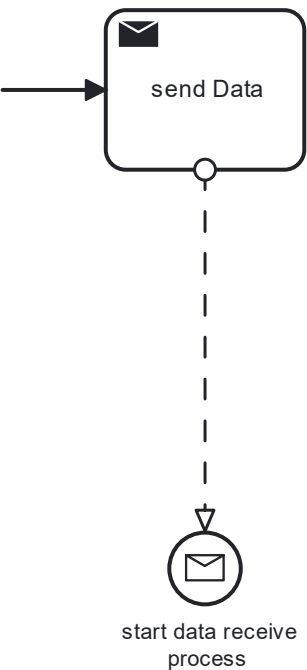


- Read from EHR, validate, encrypt, store
- Send trigger
- Download, decrypt, validate, store
- Send receipt

<https://github.com/medizininformatik-initiative/mii-process-data-transfer>



# FHIR: Task Resource



```
{
  "resourceType": "Task",
  "meta": { "profile": [
    "http://medizininformatik-initiative.de/fhir/StructureDefinition/task-data-send|1.1" ]},
  "instantiatesCanonical": "http://medizininformatik-initiative.de/bpe/Process/dataReceive|1.1",
  "status": "requested",
  "requester": { "identifier": {
    "system": "http://dsf.dev/sid/organization-identifier", "value": "sender.org"}},
  "restriction": { "recipient": [{ "identifier": {
    "system": "http://dsf.dev/sid/organization-identifier", "value": "receiver.org"}}]},
  "input": [
    { "type": { "coding": [{
      "system": "http://dsf.dev/fhir/CodeSystem/bpmn-message", "code": "message-name" }]}},
    "valueString": "dataSend" },
    { "type": { "coding": [{
      "system": "http://medizininformatik-initiative.de/fhir/CodeSystem/data-transfer",
      "code": "document-reference-location" }]}},
      "valueReference": {
        "reference": "https://dsf.sender.org/fhir/DocumentReference/600c177d-e404-42de-a773-8953b6cbef70" }}
  ]
}
```

<https://github.com/medizininformatik-initiative/mii-process-data-transfer>

# Java: Asymmetric Encryption with ECDH KEM

```
public class DemoCryptoService implements dev.dsf.bpe.v2.activity.ServiceTask
{
    @Override
    public void execute(ProcessPluginApi api, Variables variables) throws ErrorBoundaryEvent, Exception
    {
        KeyPair x448KeyPair = api.getCryptoService().createKeyPairGeneratorX448AndInitialize().generateKeyPair();
        byte[] encrypted = send(api.getCryptoService(), x448KeyPair.getPublic());
        receive(api.getCryptoService(), x448KeyPair.getPrivate(), encrypted);
    }

    private byte[] send(CryptoService cryptoService, PublicKey publicKey) throws Exception
    {
        byte[] encrypted = cryptoService.createEcDhKem().encrypt("sensitive-data".getBytes(StandardCharsets.UTF_8), publicKey);
        // store encrypted data in local DSF FHIR server, create download trigger task in remote DSF FHIR server
        return encrypted; // encapsulation, iv, encrypted data
    }

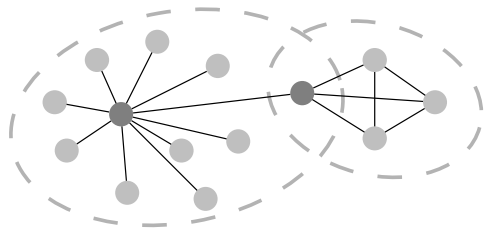
    private void receive(CryptoService cryptoService, PrivateKey privateKey, byte[] encrypted) throws Exception
    {
        // download encrypted data from remote DSF FHIR server
        byte[] decrypted = cryptoService.createEcDhKem().decrypt(encrypted, privateKey);
        assert "sensitive-data".equals(new String(decrypted, StandardCharsets.UTF_8));
    }
}
```



# Summary

---

**Extendable** with independent use case specific process plugins



**Configurable** to connect projects, organizations and research networks

**Standardized** using HL7 FHIR R4 for communication and data storage as well as BPMN 2.0 for visualization and logic



**Deployed** with 50 installations in Germany



## Additional Information and Contact

---



**dsf.dev**

- FHIR Implementation Guide
- Plugin API JavaDoc

### GECKO Institute

Prof. Dr. Fegeler

Jan Böhringer, Hauke Hund, Max Kurscheidt, Simon Schweizer

[dsf-gecko@hs-heilbronn.de](mailto:dsf-gecko@hs-heilbronn.de)



HEILBRONN UNIVERSITY  
OF APPLIED SCIENCES