OCP Code Repositories

Rev: 10/18/19 13:40

Repository	Description and Latest Commit Date
ocp-ui-ot	Latest commit 14 days ago
c2s-sof-api-ot	**Consent2Share SMART on FHIR API** The Consent2Share SMART on FHIR API (c2s-sof-api) is a Backend component of Consent2Share (C2S) Latest commit 20 days ago
c2s-sof-ui-ot	Latest commit on Aug 10
ocp-fis-ot	**Omnibus Care Plan (OCP) Service** The Omnibus Care Plan (OCP) service is a Spring Boot project provides RESTful endpoints to allow applications to publish and retrieve FHIR resources. Latest commit 24 days ago
ocp-ui-api-ot	**Omni Care Plan (OCP) User Interface API** The Omni Care Plan (OCP) User Interface API (ocp-ui-api) is a Backend For Frontend (BFF) component of ocp-u Latest commit 26 days ago
ocp-uaa-ot	**CloudFoundry User Account and Authentication (UAA) Server** The UAA is a multi-tenant identity management service, used in Cloud Foundry, but also available as a standalone OAuth2 server. Its primary role is as an OAuth2 provider, issuing tokens for client applications to use when they act on behalf of Cloud Foundry users. It can also authenticate users with their Cloud Foundry credentials and can act as an SSO service using those credentials (or others). It has endpoints for managing user accounts and for registering OAuth2 clients, as well as various other management functions. Latest commit on Aug 10
c2s-sof-ui-ot	Latest commit on Aug 10
ocp-vss-ot	**Value Set Service** The Value Set Service (VSS) is responsible for Managing sensitive categories, code systems, value sets and coded concepts. The VSS also provides a RESTful service to map coded concepts to respective sensitive categories and provide the list of all sensitive categories available in the system. Latest commit on Jun 22
ocp-pep-ot	**Policy Enforcement Point Core Service** The Policy Enforcement Point (PEP) Service is one of the core components of the Consent2Share(C2S) application. The PEP delegates the access decision to the Context Handler API, and it utilizes the Document Segmentation Service (DSS) for segmenting CCD documents according to a patient's granular consent. PEP gives the same response for both "No applicable consents" and "No documents found" cases to avoid exposing the existence of a patient's consent.

	Latest commit on Jan 3
ocp-try-policy-ot	**Try My Policy** The Try My Policy (TRY) is a service that enables patients to preview the redacted version of their uploaded clinical document based on the privacy preferences of the consent. It calls the Document Segmentation Service (DSS) to (1) segment the patient's clinical document, in the template prescribed by C-CDA-R1, C-CDA-R2, and HITSP C32, and (2) highlight the sections that will be removed in accordance to the patient's consent. Try My Policy transforms the response from DSS into a full XHTML file and provides the Base 64 encoded file in the response JSON. This service is currently utilized in the Consent2Share UI (c2s-ui) for patients to try their policies on their uploaded documents. Latest commit on Jan 3
ocp-context-handler-ot	**Context Handler** The Context Handler is a RESTful web service component of Consent2Share. It is responsible for sending XACML response context that includes authorization decisions along with applied policy obligations to Policy Enforcement Point (PEP) components. The Context Handler sends the XACML request context to the Policy Decision Point (PDP). The PDP evaluates the applicable policies either from the Patient Consent Management (PCM) or a Fast Healthcare Interoperability Resource (FHIR) server against the request context and returns the response context that includes authorization decisions along with obligations of applied policies to the Context Handler. The Context Handler sends XACML response context back to the PEP component. The PDP uses HERAS-AF, an open source XACML 2.0 implementation, for XACML evaluation, and uses either a PCM database as a local policy repository or a FHIR server to retrieve XACML policies that are generated from patients' consents. Latest commit on Mar 8
ocp-dss-ot	**Document Segmentation Service** The Document Segmentation Service (DSS) is responsible for the segmentation of the patient's sensitive health information using the privacy settings selected in the patient's consent. The segmentation process involves the following phases: 1. Document Validation: The DSS uses the Document-Validator to verify that the original document is a valid CCD document. 2. Fact Model Extraction: The DSS extracts a fact model, which is essentially based on the coded concepts in a CCD document. 3. Value Set Lookup: For every code and code system in the fact model, the DSS uses the Value Set Service (VSS) API to lookup the value set categories. The value set categories are also stored in the fact model for future use in the *Redaction* and *Tagging* phases. 4. BRMS (Business Rules Management Service) Execution: The DSS retrieves the business rules that are stored in a *JBoss Guvnor* instance and executes them with the fact model. The business rule execution response might contain directives regarding the *Tagging* phase. 5. Redaction: The DSS redacts sensitive health information for any sensitive value set categories which the patient did not consent to share in his/her consent. *NOTE: Additional Redaction Handlers are also being configured for other clean-up purposes.*

	6. Tagging: The DSS tags the document based on the business rule execution response from the *BRMS Execution* step. 7. Clean Up: The DSS removes the custom elements that were added to the document for tracing purposes. 8. Segmented Document Validation: The DSS validates the final segmented document before returning it to ensure the output of DSS is still a valid CCD document. 9. Auditing: If enabled, the DSS also audits the segmentation process using *Logback Audit* server. Latest commit on Mar 8
smart-core-ot	**SMART on FHIR Core Service** The SMART on FHIR Core Service is a Spring Boot project that provides endpoints to manage SMART App authorization process. Latest commit on Jan 3
smart-gateway-ot	**SMART on FHIR Gateway** The SMART on FHIR Gateway is a Spring Boot project that provides OAuth2 security layer and API aggregation to SMART on FHIR Core Service endpoints. Latest commit on Jan 3
ocp-edge-server-ot	**Omnibus Care Plan Edge Server (OCP-EDGE-SERVER) Service** The OCP-EDGE-SERVER Service acts as a Gate Keeper to the Outside world for OCP applications. It keeps unauthorized external requests from passing through. It uses Spring Cloud Zuul as a routing framework, which serves as an entry point to the OCP applications landscape. Zuul uses Spring Cloud Ribbon to lookup available services and routes the external request to an appropriate service instance, facilitating Dynamic Routing and Load Balancing. Latest commit on Jan 3
ocp-discovery-server-ot	**Omnibus Care Plan Discovery Server (OCP-DISCOVERY-SERVER) Service** The OCP-DISCOVERY-SERVER Service is a Spring Boot project that registers OCP related micro services and discovers them with Spring Cloud and Netflix's Eureka Services. Latest commit on Jan 3
ocp-config-server-ot	**Configuration Server** The Omnibus Care Plan Configuration Server (ocp-config-server) provides support for externalized configuration in the Omnibus Care Plan (OCP) application, including the following OCP components: • OCP UI API • OCP Edge Server • OCP FIS The Configuration Server can serve the configurations from a central Git repository on file system or a remote repository like repository on GitHub. The default configuration of this server also registers itself to OCP Discovery Server, so the other microservices can dynamically discover the Configuration Server at startup and load additional configurations. The Configuration Server is based on Spring Cloud Config project. Please see the Spring Cloud Config Documentation for details. Latest commit on Jan 3

ocp-document-validator-ot	**Document Validator Service** The Document Validator Service is responsible for validating C32, C-CDA R1.1 and C-CDA R2.1 clinical documents. It is a RESTful Web Service wrapper around MDHT (Model Driven Health Tools) libraries. It does schema validation for C32 and both schema and schematron validation for C-CDA and returns the validation results from MDHT in the response. Document Validator Service is used directly by DSS (Document Segmentation Service) to validate the document before and after the segmentation. Latest commit on Jan 3
omnibus-care-plan-ot	**omnibus-care-plan** Overarching OCP Git Repository Latest commit on Nov 28, 2018