



Проект: EXCELLCITY: Повишаване на научния и иновационен капацитет в интелигентния град чрез изграждане на връзки (ТУИНИНГ) BG-RRP-2.005-0003

АРХИТЕКТУРА ЗА ИЗГРАЖДАНЕ НА FIWARE ИНФРАСТРУКТУРА

Доклад за 3.2 РАБОТЕН ПАКЕТ 3

Проект: EXCELLCITY: Повишаване на научния и иновационен капацитет в интелигентния град чрез изграждане на връзки (туининг)

Договор №: BG-RRP-2.005-0003

Бенефициент: Университет по библиотекознание и информационни технологии (УниБИТ)

Структурата за наблюдение и докладване: Изпълнителна агенция „Програма за образование“

Автор на документа

Александър Шикаланов

Подпис _____

Дата: _____

Версия на документа

.....

Дата на версията

.....



Контрол на документа

Номер на версията	Автор	Дата на версията	Статус	Одобрено от
0.1	А. Шикаланов	13.05.2024	Първа чернова за вътрешен преглед	Доц. д-р Д. Чикуртев

Контрол на промените

Номер на променяната версия	Раздели за промяна	Обобщение на причините за промяна	Автор	Нова версия
0.1	Всичко	Вътрешен преглед		

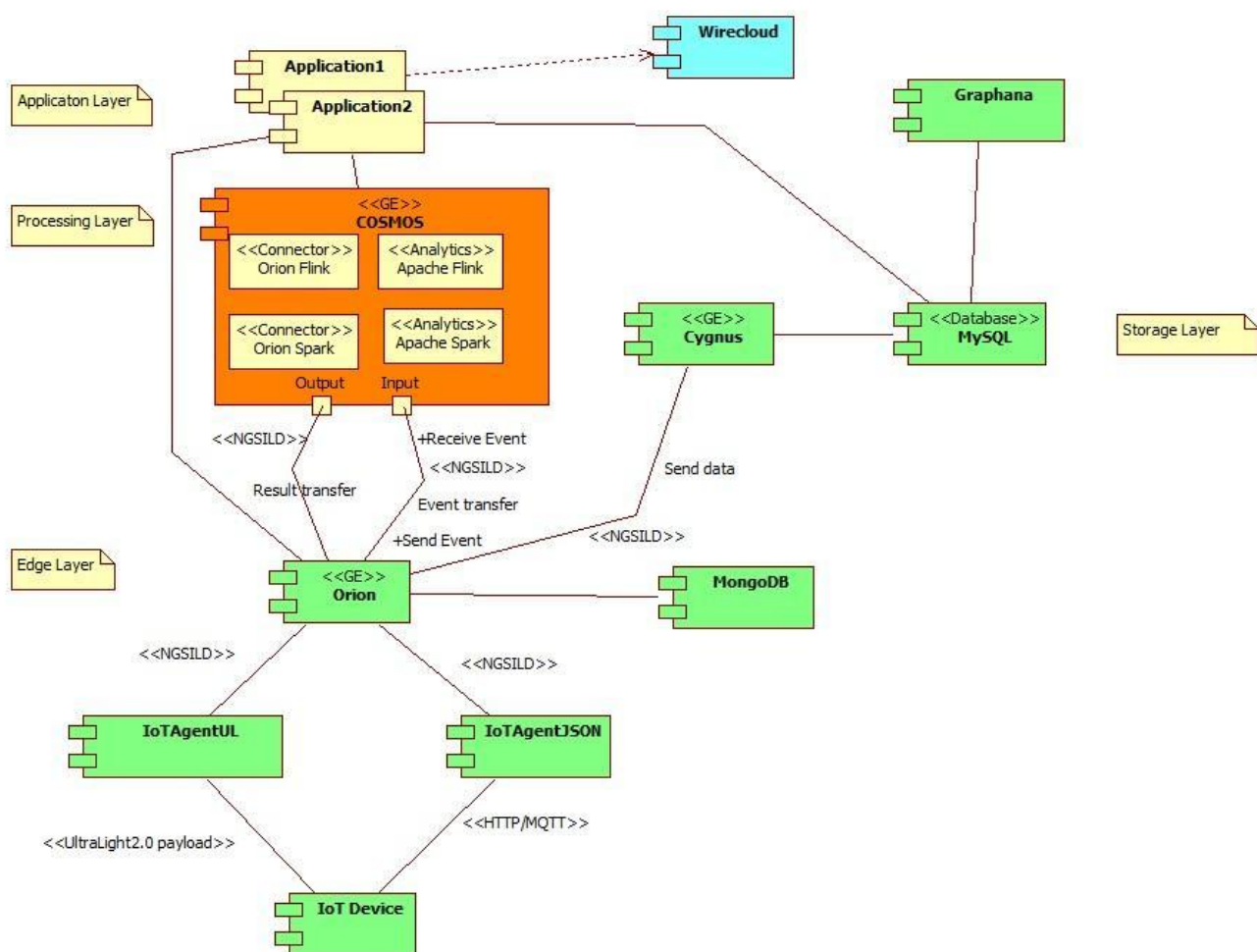
Разпределение

Всички версии на този документ се разпространяват до следните лица:

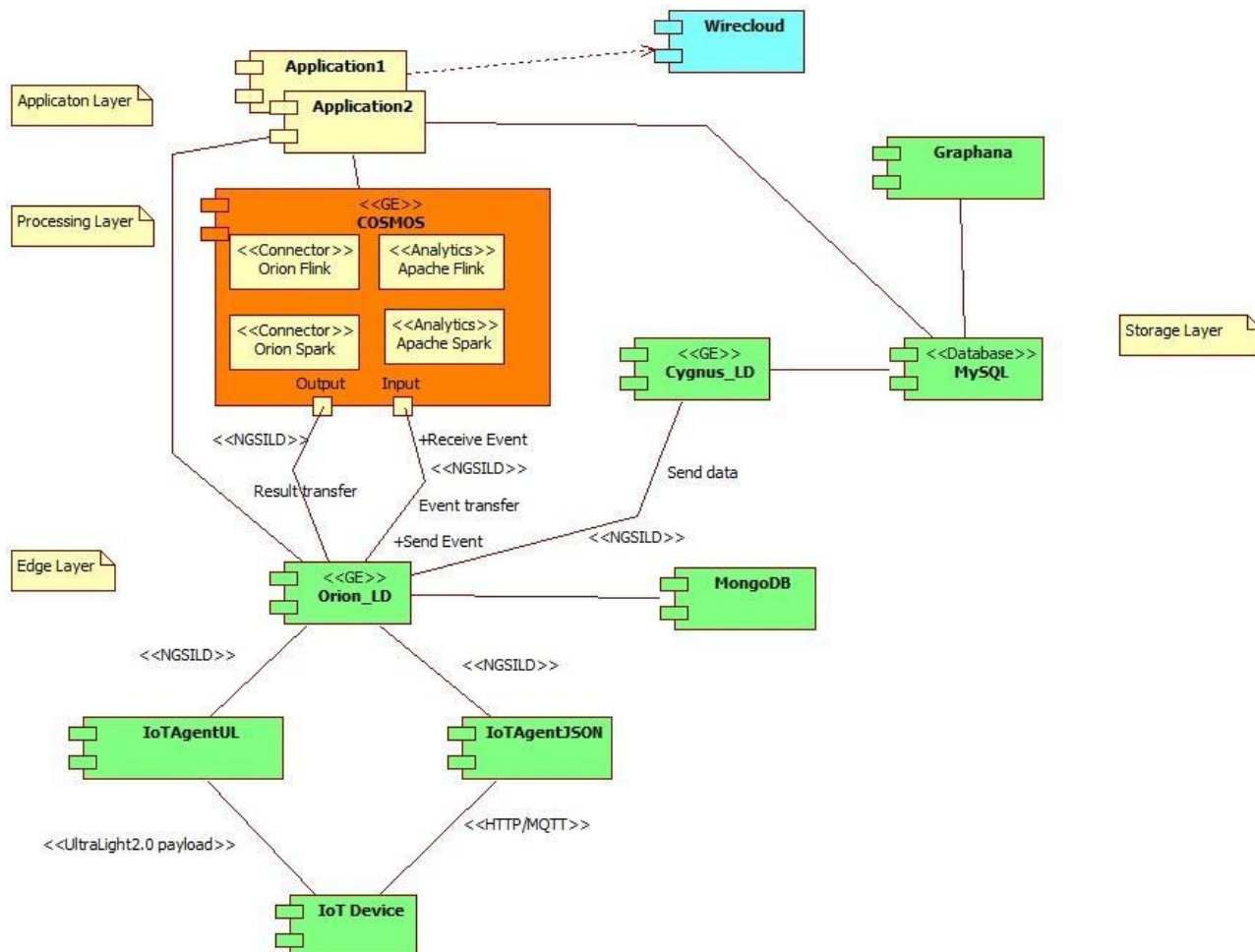
Съдържание

Референциална архитектура базирана на FIWARE	4
Референтна архитектура базирана на FIWARE - базирана на OrionLD контекстен брокер	10
Използване на Orion и Quantum leap	10
Референциална архитектура базирана на FIWARE Stellio като контекстен брокер..	11

Референциална архитектура базирана на FIWARE



Фиг. 1 Архитектура базирана на NGSrv2



Фиг. 2 Архитектура базирана на NGSI-LD

Генеричните активатори оцветени със зелен цвят са вече инсталирани и тествани.

Генеричните активатори оцветени с цианов цвят са вече инсталирани и са в процес на тестване.

Генеричните активатори оцветени с червен цвят все още не са инсталирани и трябва да се инсталират в следващ етап.

Устройствата, които не поддържат NGSIv2 или NGSI-LD могат да се свържат с Orion контекстния брокер могат да използват така наречените адаптери.

Най-разпространения адаптер е IDAS Ultralight2.0 & MQTT: Поддържа Ultralight2.0/HTTP и/или MQTT (зависи от опциите за компилиране).

Orion Context Broker (Orion, Orion_LD) е реализация на Publish/Subscribe Context Broker GE, предоставяща NGSI интерфейс. Използвайки този интерфейс, клиентите могат да извършват няколко операции:

1. Информация за контекста на заявката. Orion Context Broker съхранява контекстна информация, актуализирана от приложения, така че заявките се разрешават въз основа на тази информация. Контекстната информация се

- състои от обекти (напр. кола) и техните атрибути (напр. скоростта или местоположението на колата).
2. Актуализиране на контекстна информация, напр. изпращане на актуализации на температурата.
 3. Получаване на известие, когато настъпят промени в контекстната информация (напр. температурата се е променила)
 4. Регистриране на приложения на доставчик на контекст, напр. доставчикът на температурния сензор в помещението.

NoSQL базата данни MongoDB съхранява само последните стойности на всеки елемент данни изпратен от който и да е източник на данни свързан с FIWARE инсталацията.

Cygnus е една от възможностите за съхраняване на исторически данни, произведени чрез Context Broker. Самият Cygnus трябва да бъде свързан към магазин. В тази архитектура се използва MySQL. Не намерихме доказателство, че Cygnus работи с друга SQL или NoSQL база данни.

Релационната база данни MySQL съдържа исторически всички стойности (във времето на съществуване на източника). Таблиците съдържащи историческите стойности могат да се разглеждат като одит таблици.

Graphana е плъгин за източник на данни позволява директното използване на NGSI-LD съвместим FIWARE контекст брокер като източник на данни в Grafana. Тъй като поддържа четири различни типа заявки, т.е. времеви серии/исторически данни, текущи стойности, географски данни за визуализации на карти и данни за графики за графики на възли. Екранните снимки по-долу показват конфигурационните менюта съответно за времеви серии и заявки за карти. Филтрите за точки от данни се основават на концепциите на NGSI-LD за типове обекти, идентификатори на обекти и атрибути, така че от потребителя се изискват само основни познания за подхода за моделиране на данни FIWARE.

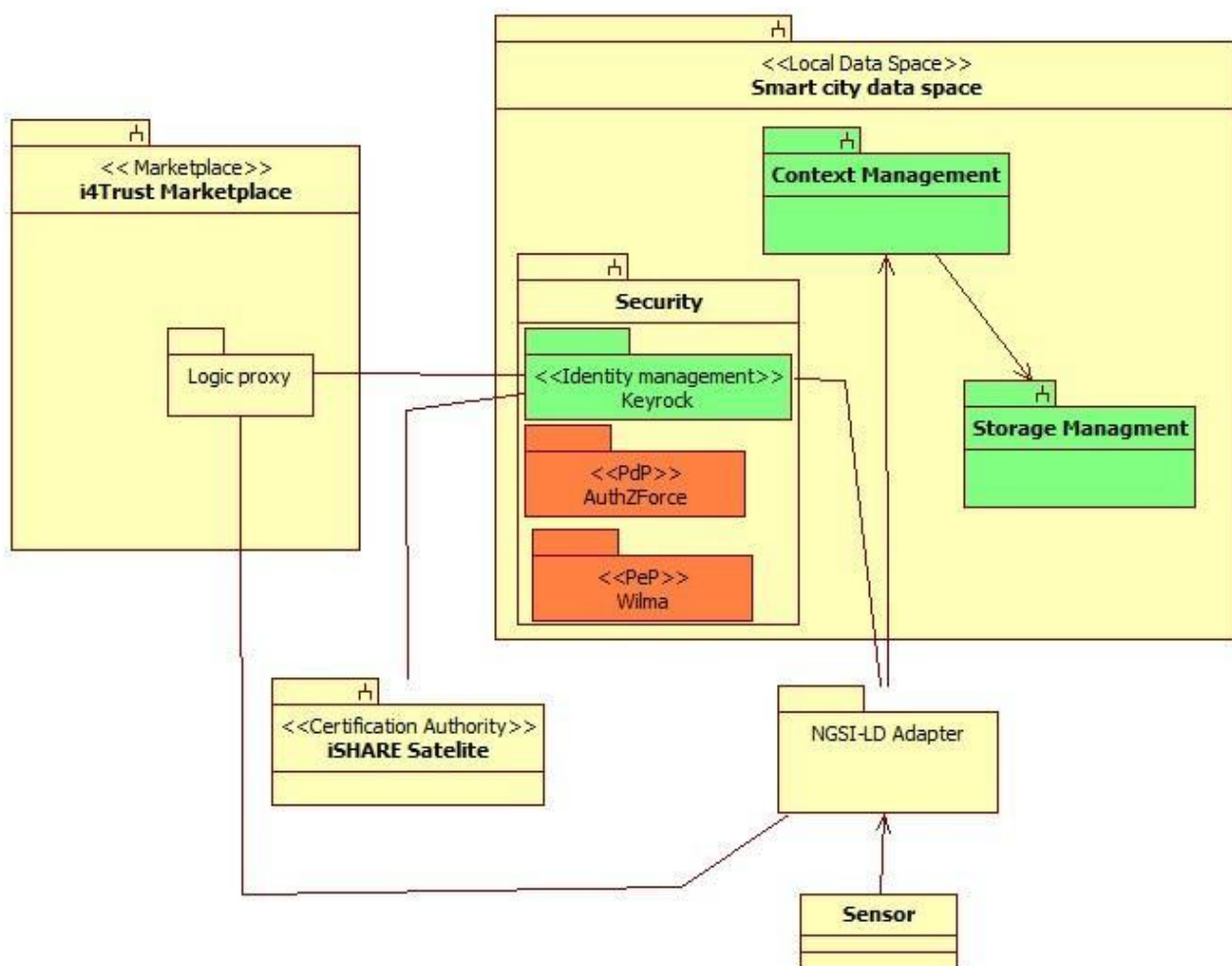
COSMOS GE се състои от четири основни компонента: Apache Flink, Apache Spark, Orion Flink connector и Orion Spark connector.

1. Apache Spark е унифицирана аналитична машина за широкомащабна (large scale) обработка на данни. Той предоставя API на високо ниво в Java, Scala, Python и R и оптимизирана машина. Тя също така поддържа богат набор от инструменти на високо ниво, включително Spark SQL за SQL и обработка на структурирани данни, API на PANDAS (Python Data Analysis Library) за работа на Spark с PANDAS API; Mllib за машинно обучение, GraphX за обработка на графики и Structured Streaming за инкрементални изчисления и обработка на потоци (скалируемост на системата).
2. Apache Flink ML е библиотека, която предоставя API за машинно обучение (ML) и инфраструктури, които опростяват изграждането на ML последователности (ML pipelines). Потребителите могат да използват ML алгоритми със стандартните ML API и допълнително да използват тези инфраструктури за изграждане на ML последователности както за обучение, така и за решаване (изводи).

И двата модула, Apache Flink и Apache Spark, са способни да извършват поточна и пакетна обработка.

WireCloud е платформа за уеб UX/UI, насочена към крайните потребители, без умения за програмиране, да създават лесно пълноценни табла за управление на приложения, изградени от графични елементи (mashups), оператори и други. Тези градивни елементи са предоставени предварително от разработчиците (и дори от други крайни потребители в случай на предварително изградени mashups) в споделен каталог. WireCloud също така предлага поддръжка за мобилни устройства, особено за таблети, така че някои от уиджетите в комбинация от приложения могат да бъдат подобрени, за да поддържат естествени функции на устройството и/или трансформирани в естествен компонент, за да подобрят тяхната производителност, когато се показват в мобилна платформа . Като пример, картата в таблото за управление на приложението, показана на следващата екранна снимка, е автоматично заменена от естествен компонент с по-добра производителност и с достъп до GPS функциите на таблета.

Структура на сигурността в FIWARE



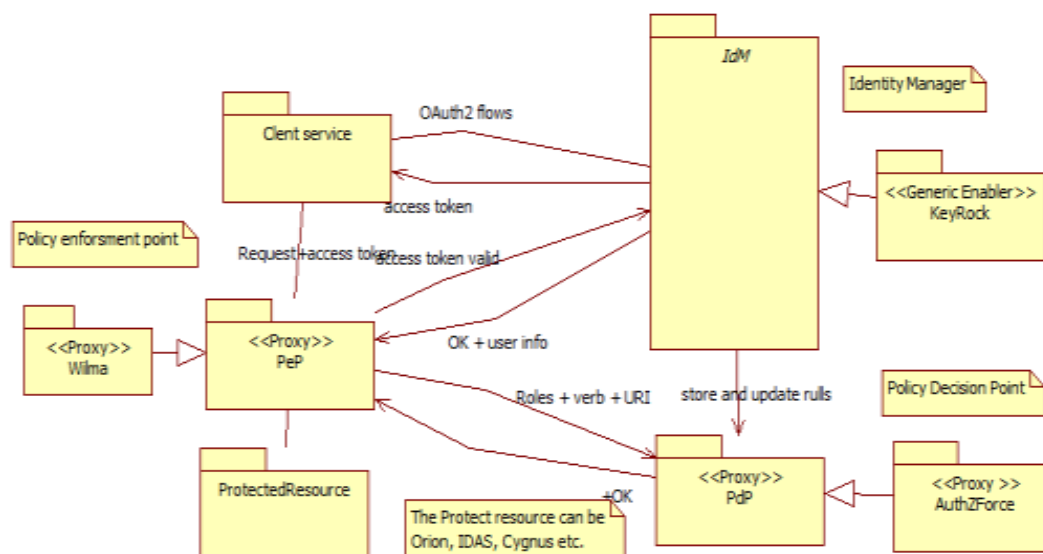
Фиг. 3 Генерични активатори свързани със сигурността в FIWARE

Във FIWARE се предлагат някои услуги и инструменти, които позволяват да се управляват удостоверяването и оторизацията в приложенията и бекенд услуги. Ако искаме да управляваме самоличността внашите приложения, без да разработваме

свои собствени механизми, можем да използваме възможността да влизане в приложението, използвайки FIWARE акаунти.

Това е възможно благодарение на протокола OAuth2 и Keyrock, компонента Identity Manager на FIWARE. По същия начин, по който обикновено влизаме в някои услуги, използвайки нашия Twitter или Facebook акаунт, нашите потребители ще използват своите FIWARE акаунти за достъп до нашата услуга. Но това е само първата стъпка. Ако нашата услуга или GE има REST API, който може да бъде достъпен от Интернет, можем да управляваме и достъпа до ресурсите. Например, можем да разрешим достъп само на потребители, които имат FIWARE акаунт.

Генеричен активатор KeyRock управлението на самоличността обхваща редица аспекти, включващи достъпа на потребителите до мрежи, услуги и приложения, включително сигурно и лично удостоверяване от потребители към устройства, мрежи и услуги, управление на оторизация и доверие, управление на потребителски профили, запазване на поверителността на лични данни, единичен Вход (SSO) за обслужване на домейни и федерация на идентичност към приложения. Менджъра на идентичност Keyrock е централният компонент, който осигурява мост между IdM системите на ниво свързаност и ниво на приложение. Благодарение на този GE можете да се създават потребители и организации във FIWARE. Може също да регистрират нови приложения, за да използва описания по-горе протокол OAuth2 и да управляват роли и разрешения за тях.



Фиг. 4 Взаимодействие на генеричните активатори при управление на сигурността.

Точка за прилагане на правилата (PEP Policy Enforcement Point). PEP Proxy се намира пред защитен ресурс и е крайна точка, открита на „добре познато“ публично място. То служи като вратар за достъп до ресурси. Потребителите или другите участници трябва да предоставят достатъчно информация на PEP проксита, за да може заявката им да успее и да премине през PEP проксита. След това PEP проксита предава заявката към реалното местоположение на самия защитен ресурс - действителното местоположение на защитения ресурс е неизвестно на външния потребител - може да се съхранява в частна мрежа зад PEP проксита или да бъде намерено на друга машина като цяло.

FIWARE Wilma е проста реализация на PEP прокси, проектиран да работи с FIWARE Keyrock Generic Enabler. Всеки път, когато потребител се опита да получи достъп до ресурса зад PEP проксита, PEP ще опише атрибутите на потребителя към Policy Decision Point (PDP), ще поиска решение за сигурност и ще наложи решението. (Разрешава или отказва). Има минимално прекъсване на достъпа за оторизирани потребители - полученият отговор е същият, както ако са имали директен достъп до защитената услуга. На неоторизираните потребители просто се връща 401 - Неупълномощен отговор.

AuthZForce е референтната реализация на Authorization PDP Generic Enabler (по-рано наричан Access Control GE). Както се изисква от спецификацията на GE, това внедряване предоставя API за получаване на решения за оторизация въз основа на политики за оторизация и заявки за оторизация от PEP. API следва стила на архитектурата REST и е в съответствие с XACML v3.0. XACML (eXtensible Access Control Markup Language) е стандарт на OASIS за формат на политика за оторизация и логика за оценка, както и за формат на заявка/отговор за решение за оторизация. Термините PDP (Policy Decision Point) и PEP (Policy Enforcement Point) са дефинирани в стандарта XACML. Този GEri играе ролята на PDP.

Насоки на следващи инсталации

Stellio – друг контекстен брокер.

Draco – друг активатор за исторически записи. Твърди се, че поддържа и други бази данни освен MySQL.

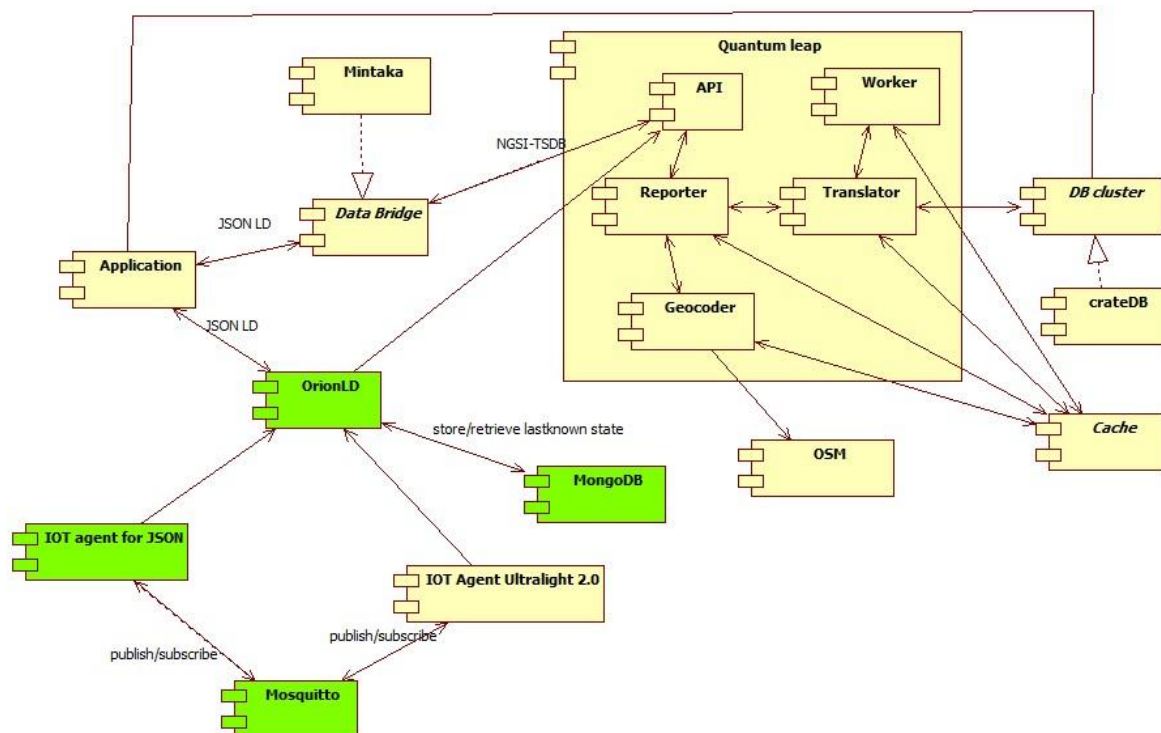
Kurento – Обработка на медиен поток в реално време.

SKAN - Публикуване на данни и монетизация.

KeyCloak – един от най-разпространените мениджъри на идентичността, възприет от FIWARE.

Референтна архитектура базирана на FIWARE - базирана на OrionLD контекстен брокер

Използване на Orion и Quantum leap



QuantumLeap е услуга за съхраняване, селектиране и извличане на пространствено-времеви данни по протоколите **NGSI v2** и **NGSI-LD** (експериментална поддръжка). **QuantumLeap** преобразува **NGSI** данните в табличен формат и ги съхранява в база данни с времеви серии, свързвайки всеки запис в базата данни с времеви индекс и, ако присъства в **NGSI** данните, местоположение на Земята. След това клиентите на REST могат да извличат **NGSI** обекти чрез филтриране на набори от обекти чрез времеви диапазони и пространствени оператори. От гледна точка на клиента, тези заявки са дефинирани върху **NGSI** обекти, за разлика от таблиците на базата данни. Въпреки това функционалността за заявки, налична чрез интерфейса REST, е доста проста и повечето сложни заявки обикновено изискват клиентите да използват директно базата данни.

Спецификацията на REST API, наречена **NGSI-TSDB**, която **QuantumLeap** поддържа, е дефинирана с цел предоставяне на независим от базата данни REST интерфейс за съхранение, заявки и извличане на времеви серии на **NGSI** обекти, които биха могли да бъдат възможно най-близо до самата спецификация на **NGSI**. По този начин **NGSI-TSDB** предоставя единен и познат (на разработчиците на FIWARE) механизъм за достъп до данни от времеви серии, което позволява внедряване на услуги като **QuantumLeap** за прозрачна поддръжка на множество backends на база данни. Всъщност понастоящем **QuantumLeap** поддържа както **CrateDB**, така и **Timescale** като backends бази данни.

Компонентът **Reporter** на **QuantumLeap** анализира и потвърждава данните, предадени с POST заявка. Освен това, ако е конфигурирано геокодиране, **Reporter** извиква компонента **Geocoder**, за да хармонизира представянето на местоположението на нотифицираните обекти, което включва търсене на географска информация в **OpenStreetMap (OSM)**.

В първия случай **Reporter** предава валидираните и хармонизирани **NGSI** обекти към компонента **Translator**. Във втория случай **Reporter** съхранява валидираните и хармонизирани **NGSI** обекти в компонента **Cache**, който действа като опашка за съобщения. Компонентът **Worker** ще прочете чакащите съобщения за обработка и ще ги предаде на компонент **Translator**.

CrateDB включва

- Анализиране на релационни, JSON, времеви серии, геопространствени, пълнотекстови и векторни данни в рамките на една система, елиминирайки необходимостта от множество бази данни.
- Напълно разпределената SQL машина за заявки, изградена върху Apache Lucene и наследяваща технологии от Elasticsearch/OpenSearch, осигурява производителни агрегации и разширени SQL функции като JOIN и CTE върху големи масиви от полуструктурирани данни.
- Индексирането в реално време автоматично индексира всички колони, включително вложени структури, докато данните се поглъщат, елиминирайки необходимостта да се притеснявате за стратегията за индексиране.
- Гъвкавата схема на данни се адаптира динамично въз основа на данните, които приемате, предлагайки безпроблемна интеграция и незабавна готовност за анализ. Колонното съхранение позволява бърза заявка за търсене и производителност на агрегиране.
- Съвместимостта на Query протокола на PostgreSQL и HTTP интерфейсът осигуряват многостранни възможности за интеграция.
- Готов за AI: Подсистемата за векторно съхранение се интегрира добре с обширна екосистема от AI/ML рамки на трета страна за усъвършенстван анализ на данни и решения, управлявани от данни.

Референциална архитектура базирана на FIWARE Stellio като контекстен брокер

Stellio се състои от 2 бизнес услуги:

- Една услуга (наречена услуга за търсене **search-service** в източника) отговаря за обработката на заявките за основните и временните API. Той е подкрепен от база данни **PostgreSQL**, разширена с разширенията **TimescaleDB** и **PostGIS**
- Една услуга (наречена абонаментна услуга в източника) отговаря за обработката на заявките за API за абонамент. Той е подкрепен от база данни **PostgreSQL**, разширена с разширението **PostGIS**

Допълва се с:

- Компонент на **API Gateway**, базиран на **Spring Cloud Gateway**, който просто препраща входящите заявки към една от услугите надолу по веригата, въз основа на пътя на заявката
- Механизъм за стрийминг на **Kafka**, който отделя комуникацията вътре в брокера (и позволява безпроблемно включване на други услуги).

