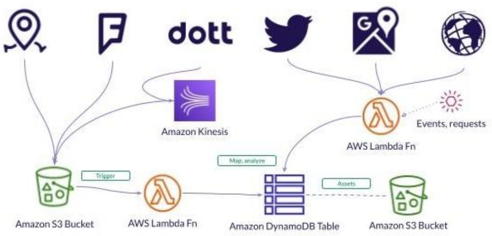
Technical Specification Double-side Page

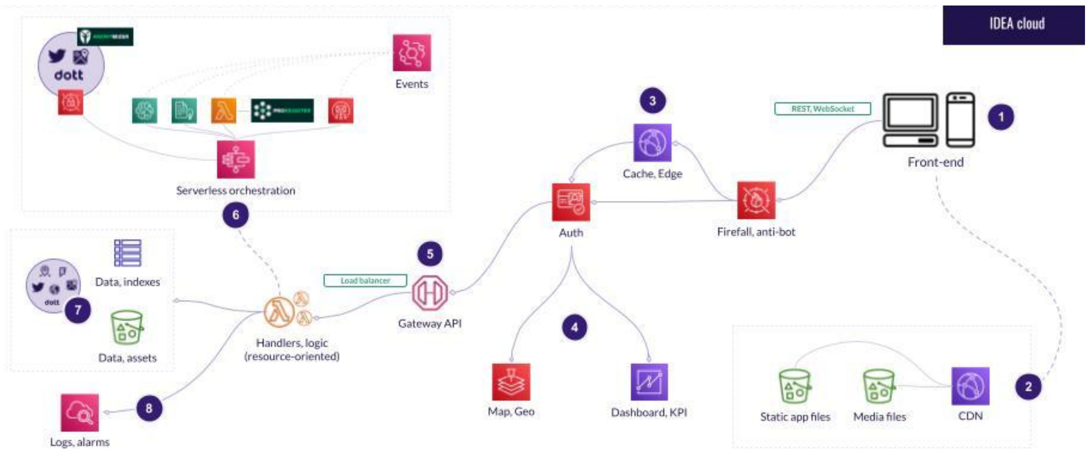
1. **TECHNICAL SCOPE:** Summarize the mock-up devised during the EXPLORE phase: how have you addressed the challenge/Theme Challenges and tackled its requirements and data. Include a diagram.

We are building **a trusted, actionable data-driven map,** for public administrations and companies, **to monitor and promote interactive experiences and their perceptions in smart communities**.

To support such an ambitious plan, we chose top-quality providers that matched our business idea. **Play&Go**, with its 10k Virtual Reality impressions collected annually; **Dott**, the fast-growing micromobility startup, with its near-real-time statistics on the turnout areas and promo codes based on 50k vehicles in Europe. Then **Twitter**, **Foursquare**, **Google Maps** and **ESRI** to access impressions, ratings and tips to elaborate through Machine Learning algorithms for extracting sentiment, knowledge and value. The **REACH toolbox** was vital in designing features to establish trust and foster interoperability and transparency with stakeholders of multiple industries.



The entire architecture is built with a **cloud-native, serverless approach** (mainly on the AWS cloud). Since their first appearances several years ago, we have invested in these technologies (for 7 years), and we can now take advantage of this incredibly **sustainable, economical, scalable** and **secure** infrastructure.



2. **ALGORITHMS, TOOLS AND CONCLUSIONS:** Detail the algorithms and tools identified to accomplish the challenge/Theme Challenges. Show clear understanding of the used REACH dataset/s and addressed challenge/Theme Challenges.

We designed our "business logic" around four pillars.

1. **Event-driven microservices**. The entire infrastructure runs on the **coordination** (via AWS Step Functions, AWS Lambda and other serverless computing tools) of powerful, modular **micro-services**, both custom and off-the-shelf, that communicate through events and event buses using the service AWS EventBridge.
2. **Location technology**. We applied everything we've learnt from previous projects to design an intelligent location technology that can support our case. We use a harmonised combination of features from Amazon Location Service powered by ESRI, Google Maps and Foursquare to provide high-quality and precise results, displayed in beautiful maps and dashboards (often powered by AWS QuickSight).
3. **Built-in trust**. Our goal is to ensure interoperability and transparency to support use cases with stakeholders from multiple industries. This desire found a match in a set of services from the **REACH toolbox**. For instance, we immediately loved the **GDPR-complaint Blockchain-as-a-Service, ProRegister**; it powers two of our major features: "***log trail***" for governance and auditing and "***log snap***" for a trusted timeline of our data's evolution. Moreover, thanks to their REST APIs, we could smoothly fit **Anonymizer** and its **comprehensive anonymisation features**, to fix data from our data providers.
4. **AI-enhanced processing**. While designing the Machine Learning stack, we analysed several papers and use cases linked to our main issues. For instance, we studied how comments from Twitter and other social networks are better understood from specific pre-trained algorithms rather than standard academic techniques. Hence, we

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951981

involved features from Amazon Comprehend and Amazon SageMaker. For example, we perform a **multi-lingual *targeted* sentiment analysis** on our contents, identifying sentiment towards individual entities inside a paragraph (instead of on the whole block). Moreover, an LDA-based learning model powers our **topic modelling components for monitoring online impressions**.

3. **SCALABILITY AND FLEXIBILITY OF THE SOLUTION:** Discuss whether the solution can truly cope with humongous and increasing datasets and how flexible it is to adapt to other related domains

We chose some of the best data providers to support our use case; with that came **a deep complexity in terms of data flow and mapping** that we had to address with equally valid tools.

For example, the **diversity of data models** oriented us on a NoSQL system, a tool that could **handle bursts of accesses** with on-demand scalability. We had high **volumes of data**, so we needed consistent performance at scale. Our solution needed to be **robust and secure**, so we required a tool with high availability (e.g. five-nine), point in time recovery, replication, encryption and fine-grained access control. Finally, we studied systems that could ensure **good interoperability with the data providers** chosen (e.g. Foursquare, Dott) while letting us **focus on our business case** rather than on configurations, updates, etc. For these reasons, we decided to build our data lake on a tandem of services that satisfied all of our requirements, namely Amazon DynamoDB and Simple Storage Service (S3).

Our **serverless design** allows us to react quickly to massive/continuous data exchanges (Amazon Kinesis) and efficiently map what we need (AWS Lambda).

4. **DATA GOVERNANCE AND LEGAL COMPLIANCE:** Describe the security level of the proposed solution, i.e. how authentication, authorization policies, encryption or other approaches are used to keep data secure. Explain how will be compliant with the current data legislations concerning security and privacy (e.g. GDPR).

All ITER IDEA's and AWS's components and services are **built on several international compliances for risk, security, quality, PII management**, etc.: *SOC 1/ISAE 3402, SOC 2, SOC 3- FISMA, DIACAP, and FedRAMP - PCI DSS - ISO 9001, ISO 27001, ISO 27017, ISO 27018 - GDPR, FIPS 140-2, and NIST 800-171.*

We adopted **a set of AWS services to maximise security and reliability**: from **web firewalls and anti-bots** (AWS WAF) to **automatically-rotating cryptographic keys and secrets** (AWS KMS & AWS Secrets Manager). Each architectural component is built with a **least-privilege permissions approach** (AWS IAM), and the whole flow is **monitored with certified auditing tools** (AWS CloudTrail). Users **authenticate securely, with MFA** and other best practices, through our service built on AWS Cognito; their data is stored in modular regions of the AWS cloud located in **Europe**.

Generally, we provide **encryption in transit and at res**t. Services have internal **data redundancy** and **fault tolerance mechanisms** for high availability and durability. Lastly, we configured **dependency-graph** automated mechanisms and bots to identify security risks in source code and libraries.

5. **QUALITY ASSURANCE AND RISK MANAGEMENT:** Describe the quality process planned for the final product. Technologically, which are the potential risks in all the phases of the project (design of the solution, development, testing, deployment...) and indicate mitigation plans to still fulfil the challenge/Theme Challenges and data provider requirements.

Building on a standard ISO 31000-like approach to risk management, we take a further step with distinctive frameworks that guide the creation of excellent solutions in the cloud. Specifically, we chose the **AWS Well-Architected tool** to continuously improve the design and development under six pillars: **operational excellence, security, reliability, performance efficiency, cost optimisation,** and **sustainability**. Moreover, we apply a specific subset of quality assurance and risk management best practices typical of a serverless environment (***Serverless Lens***).

A non-exhaustive example under the pillar of operations excellence: *"OPS 5. How do you reduce defects, ease remediation, and improve flow into production?"*. We implemented the whole set of suggested practices: *Use version control* (➜ GitHub), *Test and validate changes* (➜ both manual and automated), Use configuration management systems (➜ IAC & AWS Parameter Stores), *Use build and deployment management systems & Fully automate integration and deployment* (➜ GitHub actions), *Perform patch management* (➜ supported by specific bots that analyse dependency graphs), *Use multiple environments* (➜ "dev", "prod" and feature-specific), etc.

Moreover, to overcome availability risks, we chose data providers that could ensure stable and versioned API (or similar interchange systems). A complex challenge/solution like the one we are undertaking has required a complete analysis throughout all the pillars. In addition, **we will need continuous monitoring to set improvement and corrective actions during** the project; nevertheless, by following this demanding but comprehensive approach, we believe we are on the right track.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951981