TCO

USE OF SOFIA2

AS BACKEND

VS CUSTOM DEVELOPMENT ON A RELATIONAL DATABASE

APRIL 2015

Versión 1

1 INDICE

2 Objectives and scope of the document

This document describes the business explanation for proposing Sofia2 as backend instead of a traditional development on a relational database.

This report compares the TCO (Total Cost of Ownership) of the Sofia2 Platform and a custom development considering the initial and running costs (software, hardware and personnel).

3 What is Sofia2?

SOFIA2 can be defined as a middleware + repository able to process thousands of events per second with Big Data storage capabilities with integrated rules, multi-protocol and multi-language interfaces and all operable from a Web console.

The distinguishing characteristics of Sofia2 over other platforms are:

- Free Experimentation Environment: CloudLab
- Operation and Management of the Platform 100% web
- Semantic Approach (light)
- Independent of the communication protocol
- Modular, Extensible and customizable
- Integrated Security
- Modular
- APIs multilanguage
- Integrated API Manager
- Rules Engine and integrated CEP Engine
- Integrated Social Media Capabilities
- Reference Implementation on Open-Source Software
- Deployment On-Premise and On-Cloud
- Integrated Big Data capabilities out-of-the-box
- SW Management and centralized configuration of devices

3.1 Differential characteristics

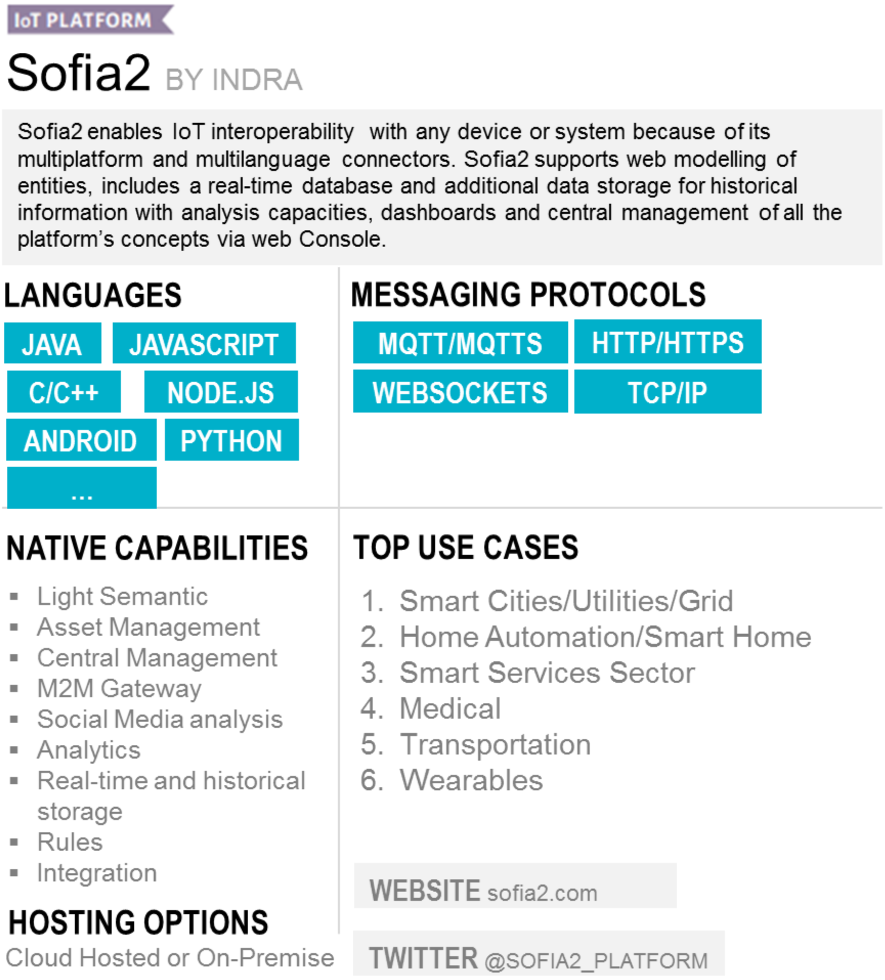
- **Free Experimentation Environment: CloudLab**
  - Within the Platform Offering there is an experimentation environment in which developers can test all the Platforms capabilities at no cost. This allows to value the Platform.
  - Additionally through CloudLab+, there is also offered an installation of a customized environment and support hours at an affordable price for pilot development to show the feasibility of a project.
- **Operation and Management of the Platform 100% web**
  - All the Platform is managed, maintained and operated from a Web Console. Included: definition of entities (ontologies), security, client creation, rules,…
  - In addition, APIs REST are offered for all the management, enabling management to integrate this into other management consoles.

- **Semantic Approach (light)**
    - The platform provides a semantic approach, allowing to use semantics when making queries, analytical processes, …
    - A light semantic approach is proposed, based on JSON that is suitable for all kinds of devices, even the less powerful ones.
    - Models are also offered for various domains like domains in the fields of Smart City, Smart Energy. These models can be extended..
    - The entities can be created from the Web console according to the domains created.
- **Independence of the communication protocol**
    - The messages exchanged between clients and the Platform are independent of the platform, out-of-the-box connectors are offered for MQTT, REST, WebSockets, WebServices and JMS.
    - Developers can create new pluggable connectors.
- **Modular, Extensible and customizable**
    - The platform is created with a modular approach which allows replacing the implementation of one module for another contemplating the interfaces (eg using SAP HANA instead of the reference implementation of MongoDB database).
    - The platform is designed to extend the project through the concept of plugin. The plugins are deployed as JARs and they allow to build new connectors, set the appropriate security model,…
    - The installation can be customized according to needs, installing only the appropriate modules, defining parameters,…
- **Integrated Security**
    - Within the Platform, security is a fundamental concept.
    - Safety is provided at the level of communication (HTTPS, MQTTS, ...) and also at the level of permissions on entities, can be private, public or entities with personalized access.
- **APIs Multilanguage**
    - Apart from REST connector that can be used easily from any language, multilanguage APIS are offered when we more advanced and efficient protocols.
    - The APIs allow to communicate more easier with the platform.
    - The Platform offeres different APIs as Java, Javascript, C/C++, Python, Android, iOS, Node.js, Arduino,…
    - All these APIs are offered under Apache license and without cost.
- **Integrated API Manager**
    - The APIS, as a mechanism of integration, is now a reality as big companies have tested like Facebook, Twitter, Google, Banks internationally,…

    - On one hand, it makes available as REST APIs the managed entities, allowing to evolve the Open Data concept to the dynamism of REST APIs and it also makes available External APIs under a RESTful approach offering a unified access.

- **Rules Engine and integrated CEP Engine**
    - The platform includes a rules engine that allows to define in a simple language and through the Web console the rules to be applied at the arrival of an event or data or a timed rule. It allows users with permissions to create in a scripting language reusable and callable new logics from the motor (eg send SMS or mail,…)
    - In addition, there is included a CEP engine that allows you to define rules where the time intervenes (eg a certain measure has not reached the system in 1 day). Clients can subscribe to the events generated by the CEP engine or serve as input to the Rules engine.
- **Integrated Social Media Capabilities**
    - Various Searches can be made from the console (profiles, timeline, groups, hashtags, trends) for the major social networks.
    - These searches will remain on the platform for analytical processes.
- **Reference Implementation (RI) on Open-Source software**
    - The RI of the Platform is based entirely on open-source software without restrictions, this allows the platform to not involve expensive licenses in its implementation.
    - In addition there are other implementations custom made so some parts are replaced by others (eg HANA instead of Mongo and Hadoop, Oracle CEP engine instead of WSO2 CEP Engine, …)
- **Deployment On-Premise y On-Cloud**
    - The Platform can be installed On-Premise and On-Cloud, either public or private.
    - Supports various operating models and various types of support based on the needs of the organization.
    - In cloud deployment can opt for a PaaS model that is charged by the assembled infrastructure or SaaS which is charged by the number of messages processed or used TB.
- **Integrated Big Data capabilities out-of-the-box**
    - The platform integrates natively a Big Data approach, so that for each entity (ontology) we can define when the data passes from the RTDB (Real Time Database) to the HDB (historical database) (RI on Hadoop).
    - The platform integrates capabilities for online queries both on the RTDB as on the HDB.
    - The Platform allows you to run analytical processes on the BDH in a simple and integrated way.

- **SW Management and centralized configuration of devices**
    - In addition to multilingual Infrastructure, a Java APIs is offered for building self-managing applications for embedded devices.
    - The SW can be loaded from the central console and the configuration for each device or group of devices.



IoT PLATFORM

Sofia2 BY INDRA

Sofia2 enables IoT interoperability with any device or system because of its multiplatform and multilanguage connectors. Sofia2 supports web modelling of entities, includes a real-time database and additional data storage for historical information with analysis capacities, dashboards and central management of all the platform's concepts via web Console.

LANGUAGES

JAVA | JAVASCRIPT | C/C++ | NODE.JS | ANDROID | PYTHON | ...

MESSAGING PROTOCOLS

MQTT/MQTTS | HTTP/HTTPS | WEBSOCKETS | TCP/IP

NATIVE CAPABILITIES

- Light Semantic
- Asset Management
- Central Management
- M2M Gateway
- Social Media analysis
- Analytics
- Real-time and historical storage
- Rules
- Integration

HOSTING OPTIONS

Cloud Hosted or On-Premise

TOP USE CASES

1. Smart Cities/Utilities/Grid
2. Home Automation/Smart Home
3. Smart Services Sector
4. Medical
5. Transportation
6. Wearables

WEBSITE sofia2.com

TWITTER @SOFIA2_PLATFORM

4 Total Cost of Ownership (TCO)

4.1 Upfront Costs

The initial costs include:

- **Initial developer effort:** Personnel costs + Developer coding required to get the application and data store working together
- **Initial administrative effort:** Personnel costs + Admins to install and configure software, cluster machines, set up sharding, etc, …
- **Software licenses**
- **Server hardware:** Servers required to run database (excludes storage). Driven primarily by the number and type of processors and RAM. Other costs include enclosures, network connectivity, cabling, and power supplies.
- **Storage hardware:** Storage required to store the data, varies depending on whether internal or shared (SAN) storage is used, the amount of storage and whether hard disk drives (HDDs) or solid state drives (SSDs) are used.

4.2 Ongoing Costs

Consistes of:

- **Ongoing developer effort:** Personnel + Coding needed to adapt data store to the customer, market and business needs.
- **Ongoing administrative effort:** Personnel + Administrative effort required to keep the data store healthy and running.
- **Software maintenance and support:** Maintenance: Upgrades and bug fixes for software + Support: On-call assistance for troubleshooting technical problems with software.
- **Hardware maintenance and support: Maintenance:** Upgrades and bug fixes for firmware and any software that may come with the hardware + Support: On-call assistance for troubleshooting technical problems with hardware.
- **Miscellaneous Deployment Costs:** Other costs associated with keeping database up and running. Includes cloud/hosting/colocation costs, bandwidth charges, electricity feed, etc.

5 TCO Using Sofia2 as Backend

Here is how using Sofia2 as Backend reduces costs making up the TCO of a system addressed as a custom development on a relational database.

5.1 Initial Developer Effort

The initial developer effort refers to the cost of time spent by the developer to ensure that the application and data warehouse work together.

In the case of a development on a relational database, the initial development effort includes tasks such as defining the data model, creating a layer of object-relational mapping (ORM), write the business logic for the application and make the presentation layer for this logic.

Sofia2 is designed to reduce development time, so that a developer can use the platform easily with any language.

With the Sofia2-Console developers can:

- Create their own entities (Ontologies in Sofia2, tables in an RDBMS, collections in MongoDB)
- Define their business rules in a simple and guided way
- Provide secure access to their entities
- CRUD access (create, read, update, delete, ...) to all these entities through any language (Java, Javascript, C, Android, ...) allowing you to develop Web MVC applications (API Java, Python, Node js), HTML5 applications (API Javascript), mobile applications (API Android, iOS, Javascript ...) or business modules (Java, Python, C, ...)
- Ability to subscribe to events, queries, rules, ... easily and independently of the messaging protocol (JMS, MQTT, AMQP, ...)
- Assisted Publishing and APIs REST website based on entities
- Integrated GIS Capabilities
- Integrated Dashboards
- Integrated reporting
- Integrated Big Data

Therefore, we can say that it is much more profitable to develop with Sofia2 that to do a custom development on a relational database.

Another important productivity advantage of Sofia2 is its design of Entities (Ontologies) oriented to dynamic documents and schemas. The way in which the platform stores data from the application corresponds with the technology and current development practices that have evolved considerably since the beginning of the industry of the relational databases 30 years ago.

Some reasons that support the productivity benefits of Sofia2 are:

- **Usability.** Sofia2 is compatible with existing development methodologies, allows developers to iterate quickly and continuously on the data model and all from a Web interface. In contrast to a traditional relational model development that imposes a strict set of constraints to develop, both in terms of data model, rule creation, changes,…
- **Data Model.** With Sofia2, the developer only has to create the data model in one place: the product´s own Web Console. In a traditional development developers need to create and maintain the data model in three places using different interfaces: the application, the database itself and the ORM layer.
- **Data Flexibility**. Unlike a RDBMS, Sofia2 allows developers to easily store polymorphic data and semi-structured and structured data in a individual data store.
- **JSON Support**. Storage in JSON, mainstay of many current applications, is done smoothly and does not require conversion. With a RDBMS, developers need to “flatten” and transform JSON to store it in relational tables, and then they have to recover the layers to make the extraction of the database.

5.2 Initial Administrative Effort

The Installation and configuration of Sofia2 is cheap and easy. The Platform consists of:

- **CDB (Configuration DataBase)**: can be any relational database. By default it runs on a MySQL embedded DB.
- **RTDB (Real Time DataBase)**: where in the RI it is MongoDB making the initial administrative effort is low, an administrator must take into account a single variable: the number of nodes in the cluster. There is only a small set of configuration settings to put the system into operation. MongoDB administrators do not need to integrate layers of cache or create custom logic horizontal partitioning to direct queries to the correct server node. Instead, the memory storage, cache and the horizontal partitioning are core capabilities of MongoDB.
- **HDB (Historical DataBase)**: It can run on MongoDB or Hadoop depending on the needs or preferences.
- **SIB + Console + Tools + API Manager + Process**: all business modules are built on Java, deployed as Web applications on any JEE application server. The bulk of the

configuration goes in the CDB so it is not necessary to create complex configuration files.

5.3 Software Licenses

Sofia2 is a platform with a free version for the open source community (Apache license) and an issue for business subscribes that can be used in different modes: On Premise or Cloud.

This version includes support in various forms (from 8x5 without SLAS to 24x7 with strict SLAS), software updates and solution to errors and some additional functions.

The commercial edition of Sofia2 is billed continuously instead of timely (ie, an annual fee per server).

5.4 Server Hardware

In general, the cost of Sofia2 servers is considerably lower than in a traditional development on a relational DB for similar workloads and availability. This applies to all components.

Sofia2 is designed to use basic hardware in scalable architectures.

Sofia2 deployments typically use basic and economic Linux servers, which cost only $3.000, even a low-power and high performance can cost just $4.000 (excluding storage).

5.5 Storage Hardware

The scalable architecture of Sofia2 significantly reduces storage costs.

Sofia2 can use local economic storage and allows to efficiently use the solid state drives (SSD).

5.6 Ongoing Developer Effort

The dynamics of current development efforts are lower than the initial development effort.

With a traditional development, the cost of making changes in the application is greater, whether they are changes in the schema of a database that is already in production (greater costs than for a database that has not yet been delivered) as in the development of logic, rules, security settings.

For example with Sofia2 it is easy for developers to add fields to entities, create new APIs, which leads to significantly lower costs and allows developers to adapt applications as the demands evolve.

5.7 Ongoing Administrative Effort

The ongoing administrative effort includes activities that keep the system in good working order (for example, updating software or hardware, performing backups and recovery times of unexpected interruptions).

Managing Sofia2 requires much less time and effort compared with traditional development.

Administration of a Sofia2 deployment primarily involves administering Linux configurations and hardware itself; so you only need to know and manage a few parameters.

5.8 Software Maintenance and Support

Sofia2 subscriptions are billed annually per core. This includes access to product support, software updates and bug fixes as well as certain functionalities that are only available in the paid edition.

5.9 Other Advantages of Sofia2

In addition to tangible cost savings, the model oriented to documents and Sofia2´s flexible schema, the platform also provides greater agility and flexibility to companies, which in turn provide advantages to generate income.

Once the Sofia2 Platform is implanted in a company, they can use the Platform (without the need to build a new infrastructure) for new deployments and to integrate data from other systems so they have them centralized in a common repository with Big Data capabilities. You can also develop Sofia2 applications with any technology and language.