1

Reference Architecture

The overall picture is the description of the system including the main building block or subsystems. Figure 1.1 shows the overall component diagram for i3-MARKET.

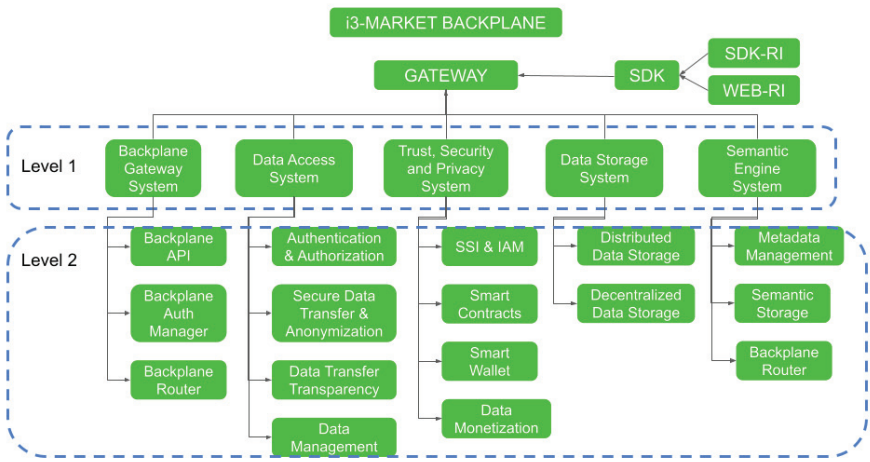

Figure 1.1 Overall system.

1.1 Level 1

Backplane gateway system:
The Backplane Gateway system is the building block in charge of offering to all participants and marketplaces access to the Backplane system. The goal of the Backplane API is therefore twofold: on one hand, it serves an integrated API endpoint for all the i3-MARKET services offered by i3-MARKET and

implemented in the respective building blocks. On the other hand, it provides secure mechanisms for preventing not allowed accesses.

In terms of public interfaces, the functionality integrated by the Backplane Gateway is exposed throughout the Backplane API.

In terms of internal connections with other i3-MARKET building blocks, the Backplane Gateway system has secure communication with the rest of the subsystems to integrate their services into the Backplane API. For this integration, any service must have a complete specification following the OpenAPI Specification 3.0.

Data access system:
The data access system is the building block in charge of allowing data consumers obtain access to the data offered by the data providers.

It exposes its functionality, publicly, through the secure data access API.

This data access system is securely linked with the Backplane API in order to guarantee two main issues.

Ensuring all the involved stakeholders have signed the required contracts and monitoring the quantities of exchanged data assets for the token-based monetization service.

Trust, security, and privacy system:
The trust, security, and privacy (TSP) is the building block in charge of providing the self-sovereign identity, access management, contracting, consents, accounting, and payments blockchain-based solutions managed in the i3-MARKET system in order to guarantee the desired levels of trust, security, and privacy for federated data markets.

The TSP system exposes its functionally, publicly, through the Backplane API.

In terms of dependencies with other existing building blocks, the TSP interacts with the decentralized ledger of the data storage system and with the data access system for the monetization of the data assets.

Semantic engine system:
The semantic engine system is the building block in charge of providing the needed semantic data models for making possible the consumers and applications understand the meaning of the data exchanged between different stakeholders. Apart from that, the semantic engine will allow the participants to take advantage of this semantic data model by means of providing a metadata management in charge of registering, offering, and performing queries for discovery purposes.

All this metadata management and query functionality is exposed, publicly, through the Backplane API.

In terms of dependencies with other building blocks, the semantic engine mainly interacts with the storage system for storing the offering descriptions.

Data storage system:
The data storage system is the building block in charge of storing common data shared across all participant instances.

It interacts with mostly all main building blocks, especially with the semantic engine system for performing the synchronization between semantic repositories and distributed storage and with the trust, security, and privacy for instantiating and executing smart contracts in the blockchain-based decentralized storage.

1.2 Level 2

Backplane gateway system – general description:
The Backplane Gateway has two main purposes:

Single entrypoint:

The Backplane offers a single set of endpoints, allowing clients to interact with all the services offered by the i3-MARKET project through a single API. This allows the whole system to have a modular and distributed architecture, while providing the ease of use of a single common interface.

Auth management:
All authentication and authorization flows are centralized and managed by the Backplane Gateway so that the subsystems are protected without them needing to handle their own auth flows. The actual authentication/authorization is delegated to specialized subsystems. This centralization also allows the support of several auth flows (and the addition of new ones) transparent to the several subsystems.

Besides these main purposes, the presence of the Backplane Gateway provides several benefits.

It allows the subsystems to be isolated and not exposed publicly so that they can only be accessed by the Backplane or other subsystems.External connections are all handled by the Backplane, making connection security and encryption simpler and more straightforward.The nature of the Backplane functionalities makes it easily scalable and replicable, making the addition of new Backplane instances transparent for both the subsystems and the clients. The Backplane Gateway is shown in Figure 1.2.

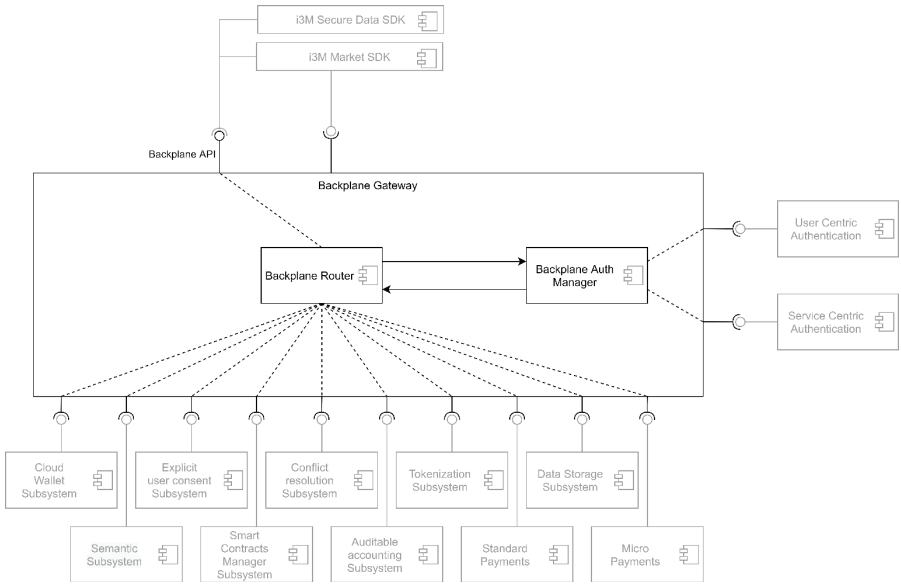

Figure 1.2 The Backplane Gateway component diagram.

Inner building blocks:
The Backplane Gateway consists of three main components.

Backplane API:
The Backplane API is the set of endpoints exposed by the Gateway. It comprises all the publicly available endpoints of the subsystems integrated with the Backplane as well as a few other endpoints, belonging to the Backplane itself, used in the authentication/authorization flows.

The API follows the OpenAPI Specification 3.0, and the endpoints corresponding to each subsystem are generated automatically based on the subsystem's own OpenAPI specification.

Backplane auth manager:
The Backplane auth manager is responsible for handling the authentication and the authorization required for the endpoints of the different subsystems. The actual auth processes are delegated to the corresponding subsystem (user/service-centric authentication subsystem), depending on the requirements of the endpoint and client.

Backplane router:
The Backplane router is the component of the Backplane responsible for the forwarding of the incoming requests to the several subsystems. It also checks

whether the endpoint requires authentication/authorization, invoking the auth manager if it does.

Data access system – general description:
The data access system consists of four main subsystems for implementing the transfer of data. Figure 1.3 shows the subsystems.

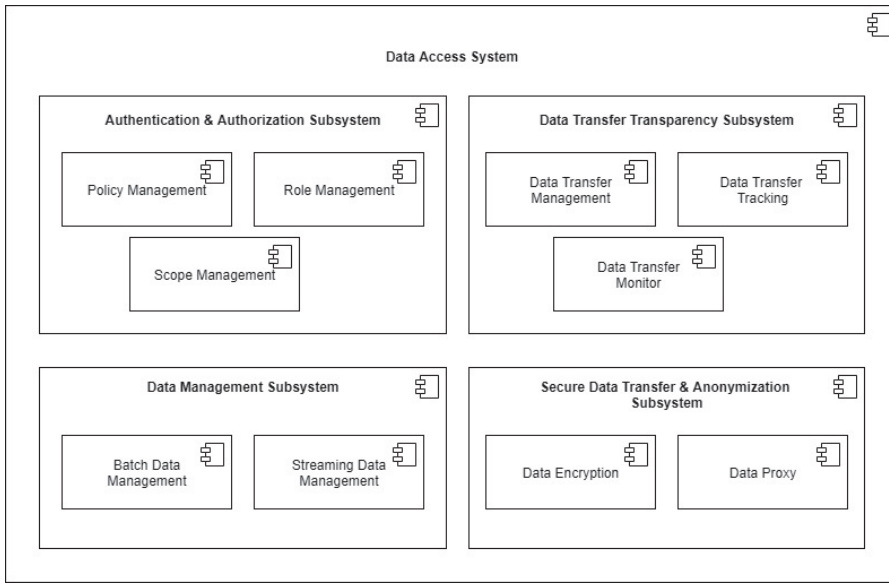

Figure 1.3 Data access component diagram.

Inner building blocks:

Authentication and authorization:
The data access system will assist i3-MARKET with the following capabilities for the exchanged data assets:

- Authentication: Verifies the identity of the user against the i3-MARKET Backplane. 7
- Authorization: Verifies the permissions the authenticated user has in the i3-MARKET platform allowing to perform authorized actions and grants access to resources.

Authentication and authorization subsystem has the following scope management:

- Policy management: Policy is a set of rules that defines how to protect the assets in order to provide trust, security, and privacy. The policy

management component is in charge of enforcing the rule set provided by i3-MARKET Backplane inside of the data access system. The responsibilities of the policy management module are:

  - intercept access attempts;
  - check attempt against rule set;
  - grant access to permitted assets.

- Role management: A role is a set of policies attached to an entity in order to define the access that entity has within the i3-MARKET data access system. The role management component is in charge of fetching the list of policies and verifying them against the data access system. The responsibilities of the role management module are:

  - get the list of policies associated with role from Backplane;
  - verify role access by invoking policy management;
  - allow or deny functionalities.

Secure data transfer and anonymization:
The secure data transfer and anonymization subsystem has the following components:

- Data encryption: The responsibilities of the data encryption module are:

  - key generation and exchange;
  - transfer data in an encrypted way between endpoints;
  - decrypt data on the consumer side.

- Proxy: The proxy needs to be used when the identity of the data provider needs to be hidden. This feature is optional; there is no need to implement it if there is no specific requirement referring to the anonymity of the data provider. The responsibilities of the proxy module are:

  - activate the proxy;
  - configure the parameters to hide the identity;
  - data transfer goes through the proxy.

Data transfer transparency:
The data transfer transparency subsystem has the following components:

- Data transfer management: This component is responsible for the management of the connection between the provider and the consumer and implements the following functionalities:

  - initialize the connection;
  - resume the connection;
  - finalize the connection.
- Data transfer tracking: This component implements the following operation:
  - measure the amount of transferred data.
- Data transfer monitor: The information about how much data was transferred, when the data transfer was initiated, and when it was completed is monitored, and the following operations are triggered:
  - inform the i3-MARKET Backplane that the data transfer was performed and reports how much data was transferred;
  - invoke the linked smart contract.

Data management:
Two methods for data transfer are supported by data access API, which are supported by the following modules:

- Batch data transfer management: One-time data transfer for one chunk of data in a session with the following methods:
  - request data;
  - transfer data.
- Data stream management: Continuous transfer of data based on a subscription, e.g., publish/subscribe mechanism:
  - subscribe to an offering;
  - trigger data transfer – on the producer side;
  - get data – on the consumer side;
  - unsubscribe.

Trust, security, and privacy system – general description:
One of the pillars of the i3-MARKET Backplane is the "trust, security, and privacy" system, which leverages the blockchain technologies to ensure trust, security, and privacy by design, providing the following building blocks as shown in Figure 1.4:

- an identity and access management system based on decentralized/self-sovereign identity and Verifiable Credentials;
- smart wallets with different levels of security (Cloud/HW Wallet);

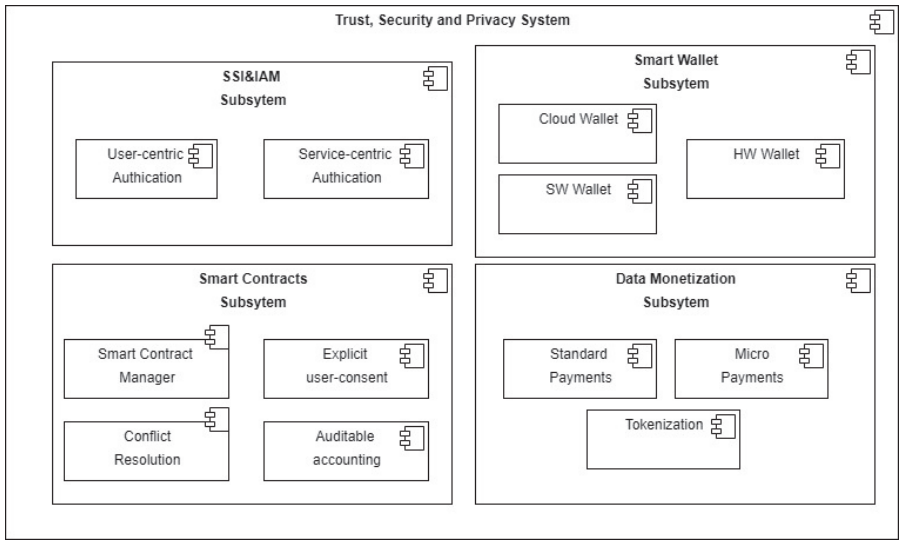

Figure 1.4 Trust, security, and privacy component diagram.

- smart contracts to record, operate, and manage in a trusted and transparent way the agreements between the different stakeholders and particularly the explicit user consent of the data owner;
- a data monetization system based on crypto currency for secure, trusted, and cost-effective peer-to-peer payments.

Inner building blocks:

SSI & IAM:
The SSI&IAM building block provides an authentication and authorization mechanism to access i3-MARKET Backplane and stakeholder resources.

The user-centric authentication component implements the self-sovereign identity paradigm ensuring that:

- identity and personal data are stored with the user;
- claims and attestations can be issued and verified between users and trusted parties;
- users selectively grant access to data, and data only needs to be verified a single time.

The service-centric authentication component makes the data marketplaces of the network able to provide their users a distributed identity they own and can use with other stakeholders.

Both the authentication mechanisms follow the OpenID Connect standard to allow wide commercial acceptance.

Smart contracts:
The Smart Contract Manager enforces certain contractual parameters of the data sharing agreement between a data provider and a consumer using predefined smart contracts, which are based on the legal agreements. The Smart Contract Manager component incorporates the conflict resolution, the explicit user-consent, and the auditable accounting component.

Smart wallet:
i3-MARKET wallets are key components that enable interaction between the different stakeholders and services in the i3-MARKET ecosystem. A wallet just stores and uses cryptographic material that, in i3-MARKET, are used to achieve the following features: authentication and authorization (by proving ownership of DIDs and Verifiable Credentials), and non-repudiation of data exchange (by digitally signing different operations).

The smart wallet building block is designed to be secure and user friendly and in a way that existing technologies can be easily added as i3-MARKET-enabled wallets. In this project, we are going to integrate three types of wallets:

- HW wallet: Hardware wallets are in charge of storing the user's private keys using a physical device as storage. It performs cryptographic operations to reduce the key exposition. This specific wallet satisfies the highest security policies since it is based on 'something you have' and you are the owner of the data.
- SW wallet: Software wallets store the user's private keys on the storage of a general-purpose device (e.g., computer or smartphone). It combines the security polices of 'something you have' and 'something you know'. Despite no specific hardware is needed, the loss of the device might imply losing the keys if no backups are made.
- Cloud wallet: Cloud wallets store the user private keys on secure cloud databases. Even though it has less strict security policies, it can offer much more functionalities than the other wallet subsystems, such as easier access and simpler key recovery.

Data monetization:
The data monetization building block provides a crypto currency solution that allows instant currency exchange among the participating data spaces/marketplaces, and also supports full audibility of all transactions. This

is vital for a fully decentralized solution, as it provides the basis for building trust in the federation backplane. The payment mechanism shall support micro-payments and requires minimal cost.

The standard payment component permits in advance an *a posteriori* payment for a specific dataset or piece of data with traditional payment systems, ensuring trust, security, and full auditability of data transfers through an *ad-hoc* non-repudiable protocol.

The tokenization component provides the creation of a crypto token for instant currency exchange among the participating data spaces/marketplaces ensuring the full auditability of payment transactions provided by the blockchain technology.

The micro payment component provides a mechanism for reducing the cost of crypto payment transactions especially for small amounts of tokens.

Semantic engine system – general description:
One of the core pillars in i3-MARKET is the semantic engine, which plays an important role in terms of registration and querying the offerings in a distributed and interoperable way. Semantic engine exposes different API endpoints for various tasks, for example, registration and querying as shown in Figure 1.5.

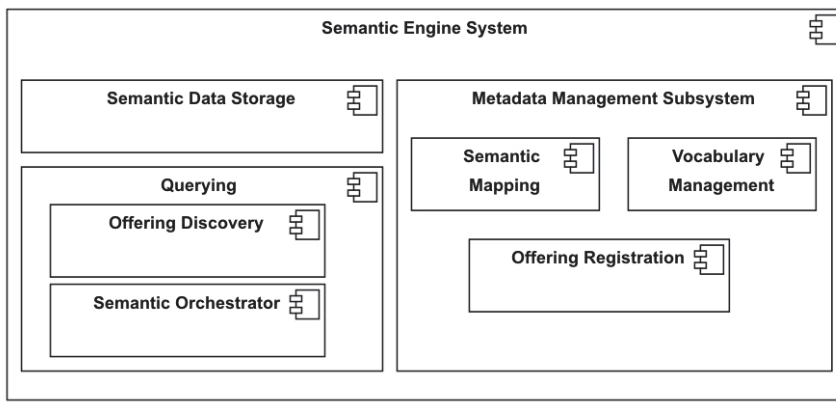

Figure 1.5 Semantic engine component diagram.

Inner building blocks:

Metadata management:
The metadata management subsystem encompasses three components:

- Offering registration: This component allows users to register offerings. Semantic engine exposes different endpoints for offering registration.

Examples are: (i) register data provider, (ii) register offering of a data provider, and (iii) update offerings.
- Semantic mapping: This component does semantic mappings and transforms (JSON to RDF) data received from API endpoints.
- Vocabulary management: This component keeps and manages all the vocabularies, defined as i3-semantic model, used in different components of the semantic engine.

Semantic storage:
This component communicates with the RDF triple store and pushes and retrieves data from that store.

In i3-MARKET, open-source virtuoso was tested as a triple store, which allows us to store RDF data and query using SPARQL query language. In general, triple stores are used to management tool for metadata, using semantic web query language (SPARQL) for accessing the information, which allows us to store RDF data and query using. At the same time, MongoDB is used to store metadata together with data descriptions and uses MongoDB query language. Figure 1.6 shows the data storage system.

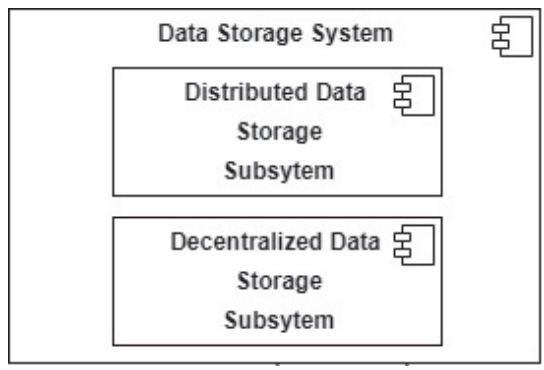

Figure 1.6 Data storage system component diagram.

Querying:
Two main components will be provided:

- Offering discovery: This component allows users, either data provider or consumers, to query already registered offerings.
- Semantic orchestrator: The role of this component is to manage synchronization with the distributed data storage component and the query processing, for instance, local or distributed query.

Data storage system – general description:

Inner building blocks:
The storage system consists of two main subsystems for implementing, respectively, the decentralized storage and distributed storage features, as shown in Figure 1.7. The subsystems are, at least in the initial architecture, relatively independent of other systems and, also, independent of each other.

Decentralized storage:
The diagram of decentralized storage subsystem is shown in Figure 1.7. The decentralized storage subsystem is implemented as a blockchain-based distributed ledger network. The software implementation is Hyperledger BESU in a permissioned setup using IBFT 2.0 consensus [10]. Hyperledger BESU uses internally an embedded RocksDB instance for storing linked blocks (the journal of transaction) and world state (the ledger). Hyperledger BESU can instantiate and execute smart contracts for supporting the use cases of the i3-MARKET framework.

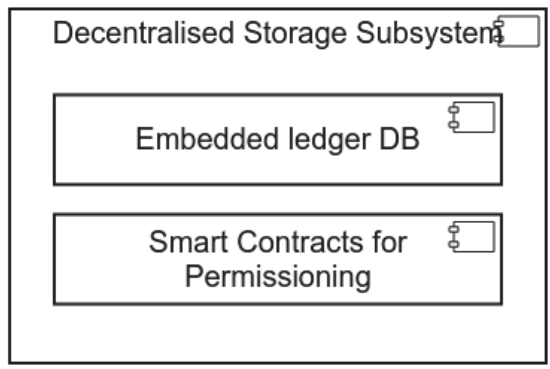

Figure 1.7 Decentralized storage component diagram.

Distributed storage:
The diagram of the distributed storage subsystem is shown in Figure 1.8. The subsystem consists of a distributed cluster of database nodes and an optional interface layer (will not be implemented for V1). The database provides an SQL interface to other i3-MARKET framework components. The software implementation of the internal database is CockroachDB that can be accessed via PostgreSQL-compatible wire protocol for which a large number of client libraries exist for different languages and platforms. CockroachDB

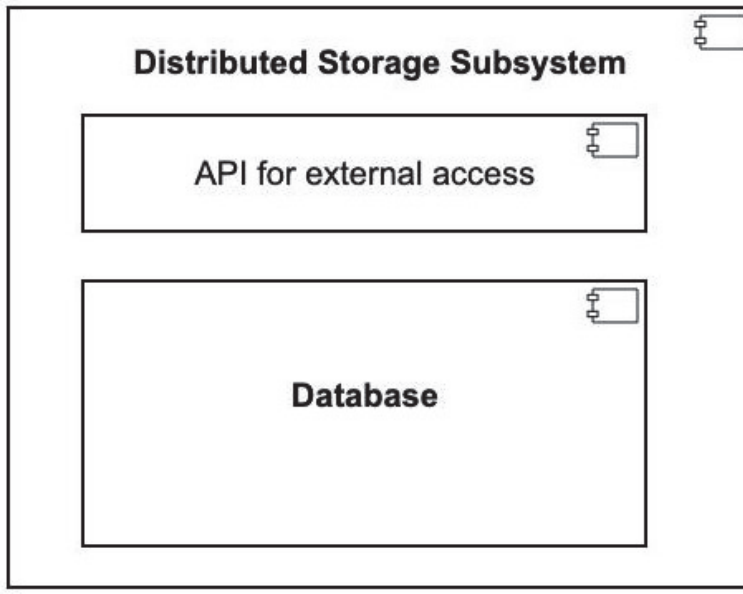

Figure 1.8 Distributed storage component diagram.

is a highly scalable and resilient distributed database. Only secure access to the database will be enabled; hence, all clients need to use private keys and valid certificates to access the database.

Deployment view:
The deployment or physical view "describes the mapping(s) of the software onto the hardware and reflects its distributed aspect"; the i3-MARKET deployment view is depicted in Figure 1.9. Four nodes constituted the i3-MARKET R1 cluster. On each node, it will be deployed a Backplane Gateway System and an instance of all the rest i3-MARKET main building blocks (trust, security, and privacy system, storage system, and data access system) giving backend support to the Backplane Gateway System. In addition to that, node 4 will host all the components related to the Semantic Engine Building Block in the form of free Open Source Software Tools for SMEs, developers, and large industries building/enhancing their data marketplaces. Figure 1.9 shows the proposed deployment as explained.

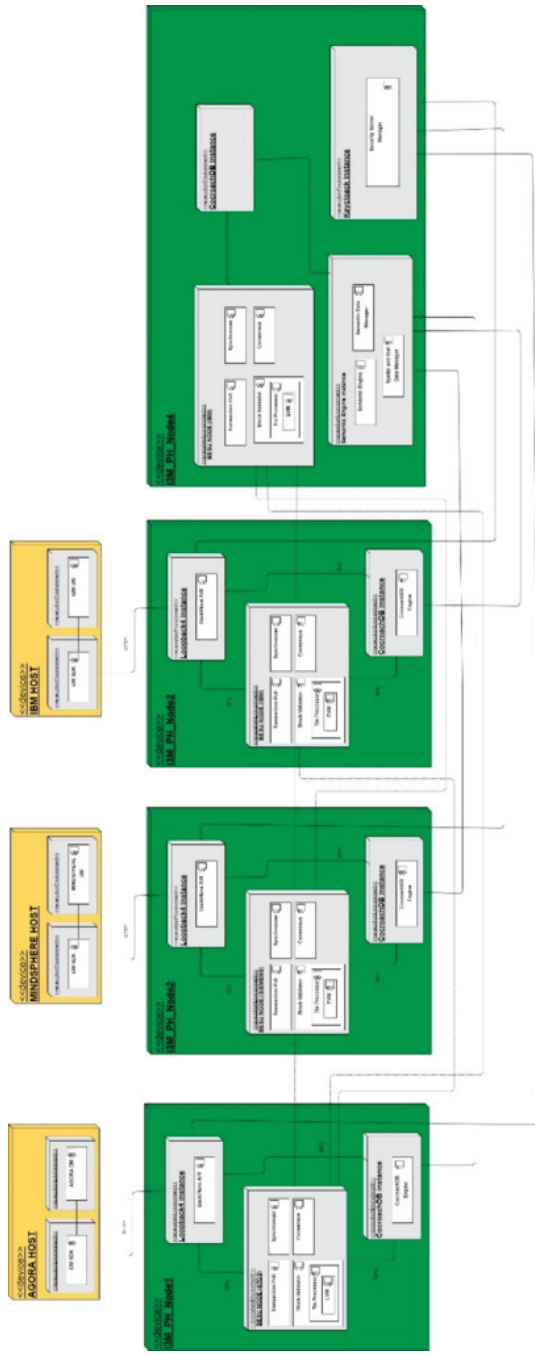

Figure 1.9 i3-MARKET deployment view.