Data storage, sharing & security.

ESME – INGE1 INTERNATIONAL TRACK

zakaria@elbazi.co

Elasticsearch/Store the data:

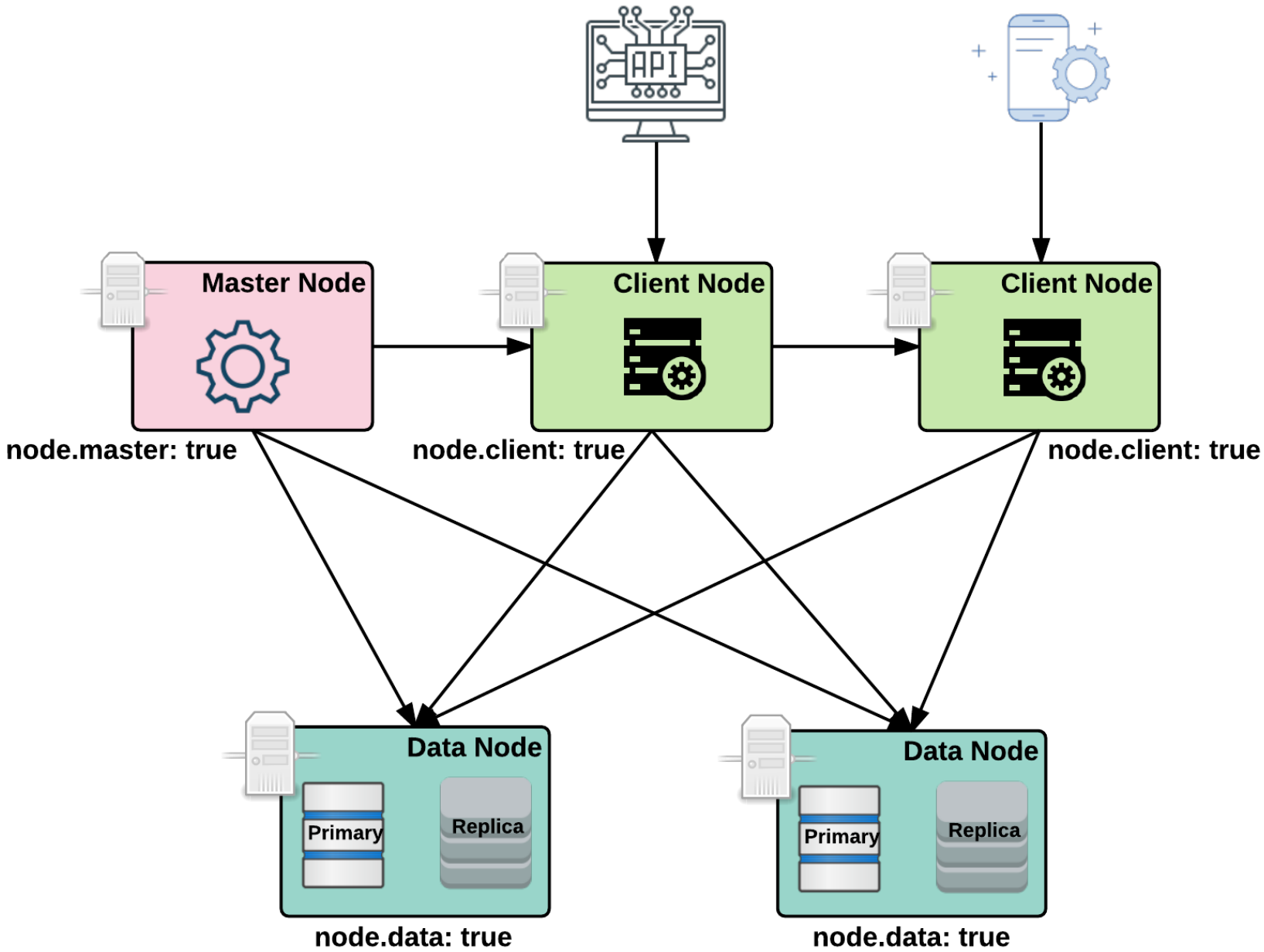
- **Elasticsearch** is a fast search engine that provides a lot of features that allow you to use it for data storage and data analysis.
- **Elasticsearch** is a document-based storage system**:**

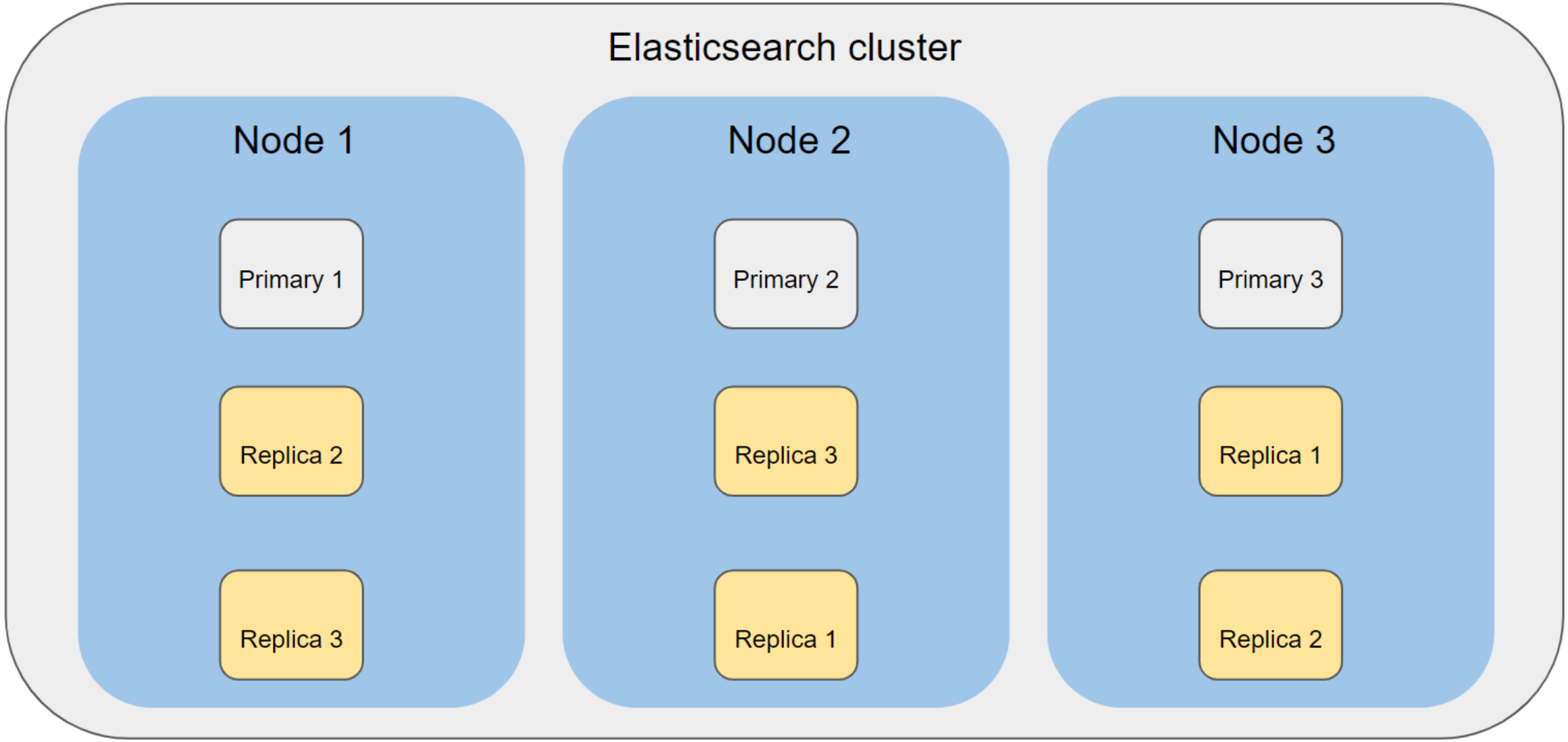
```
{
        “id”:1,
        “timestamp”:123456789,
        “temperature”:12
}
```

ESME sudria

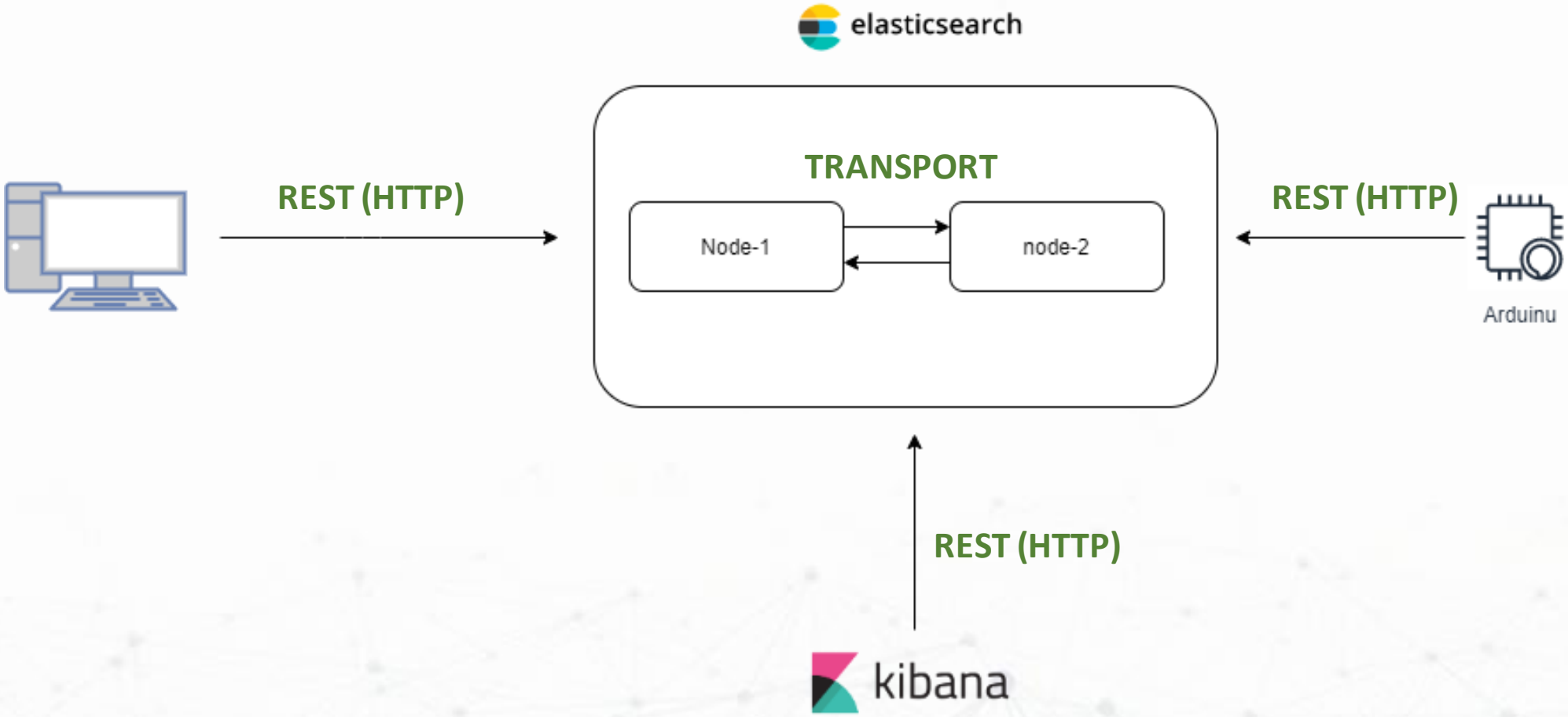
Elasticsearch/Store the data:

- **Elasticsearch** is a distributed system that's designed for **High availability (no single point of failure), scalability (capable of handling more data) and fault tolerance.**
- The primary way to interact with it to index/search or to configure it is with json-based **REST API** over **HTTP/HTTPS.**
- **Basic concepts :**
  - **Cluster :** one or more servers (nodes) that work together to store (index) data and making it searchable (we can have one node or multiple nodes clusters).
  - **Node** : a single server (physical or virtual) that stores a part (in the case of multi-node cluster) or all the data (single node cluster).

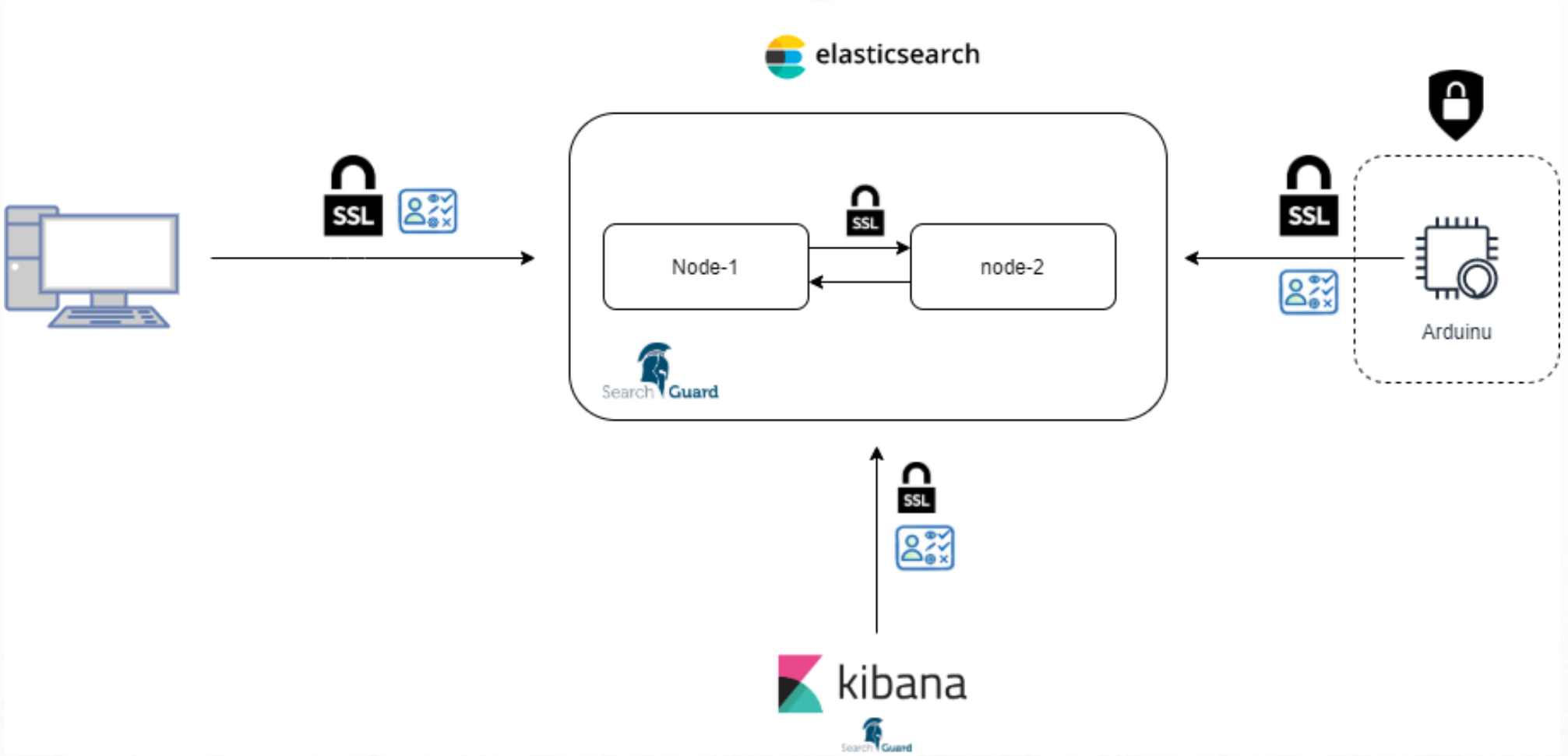

API
Master Node
Client Node
Client Node
node.master: true
node.client: true
node.client: true
Data Node
Primary
Replica
Data Node
Primary
Replica
node.data: true
node.data: true


Elasticsearch cluster
Node 1
Primary 1
Replica 2
Replica 3
Node 2
Primary 2
Replica 3
Replica 1
Node 3
Primary 3
Replica 1
Replica 2

Elasticsearch in our use case:



Elasticsearch in our use case (Security):



ESME sudria

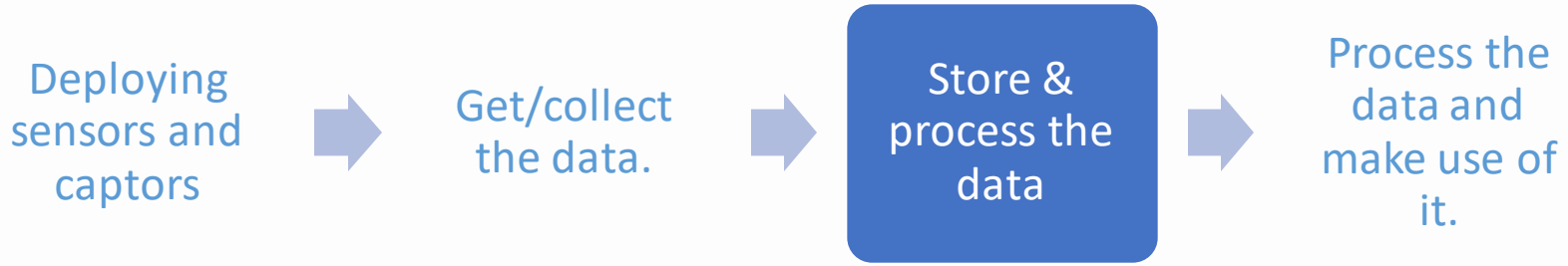
Elasticsearch in our use case (Security):

- Make the IoT devices inaccessible publicly or in an exposed network.
- Securing access to the Elasticsearch index (REST & transport):
  - Securing the communications between the user and the cluster
  - Securing inter-node communications (multi-nodes cluster)
  - Implementing **Role & permissions-based access control** to the cluster and the dashboards in Kibana.

Elasticsearch in our use case (Security with SearchGuard):

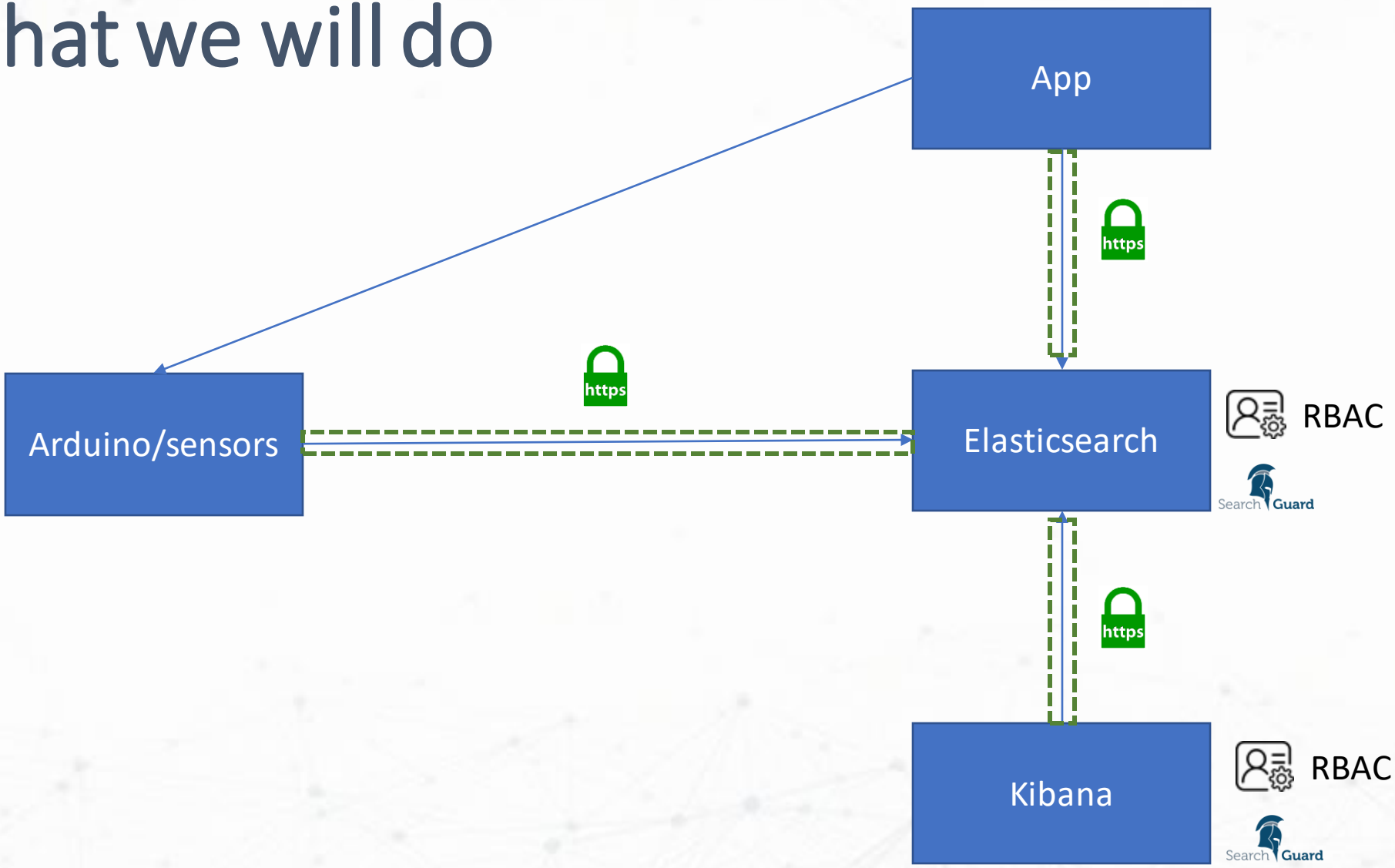
- Search Guard uses TLS on transport and REST layer:
  - **Data encryption**: No one can spy on data !
  - **Data integrity**: No one can alter the data or change it
  - **Cluster integrity**: Only trusted nodes can join the cluster
- TLS on transport layer:
  - Protects data traveling between the nodes
- TLS on REST layer:
  - **Adds HTTPS** : no one can “sniff”/see the data while transit !

Points of failure: (security by design)



- IoT devices may be exposed and publicly accessible.
- Data integrity
  - A sensor getting wrong measures for example.
  - Someone altering the data during the transfer.
- **Data storage device or data base are publicly exposed and available.**
  - **Data integrity and availability risks.**
- Applications are not enough secured (again data integrity issues).

What we will do



ESME sudria

Labs :

Elbazi.co/esme -> hands on elasticsearch.