IoT Security

Information security

- Confidentiality
  - Data accessed just by **permitted** users
- Integrity
  - **Not tampered** by not permitted users
- Availability
  - System to **access** data, from authorized user
- Overflow (flooding), Spoofing (impersonate), man-in-the-middle (listen), malware (intrusion)

Web Application security

- Application and service exposed to user via HTTP(!!!)/HTTPS
  - Communication security
- Security issues:
  - DDOS, BotNet, Post DoS (flooding)
  - SQL injection
  - Web Application Session hijack
  - Html & Js injection
- Mobile App Targets:
  - Data
  - Identity
  - Availability
- Attack based on SMS and MMS
  - + Jailbreaking

Tools for a security approach (1)

- Encryption (symmetric + asymmetric keys)
- Digital Signature
- Digital Certificate

- HTTPS (SSL/TLS)
- Authentication protocols (basic, oauth2, openIdconnect)
- JWT token, SAML, LDAP, Identity Providers (Keycloak)

Tools for a security approach (2)

- Symmetric cipher (use same unique key), fast
  - AES 128, AES 192, AES 256
  - RC4, RC5, RC6
  - DES
- Asymmetric cipher (two related keys), slow
  - RSA
  - DSA, Elliptic CURVE
  - PKCS

Tools for a security approach (3)

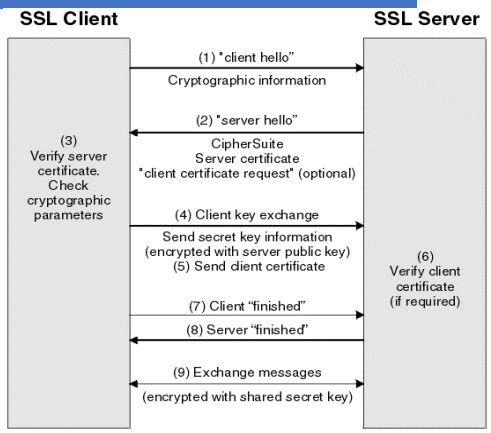
- Protection of a message
  - PubKey to **encrypt** a MSG that just the target can understand via PrivKey
- Authentication, Non repudiation, integrity
  - PrivKey to **digital sign** a message that everybody can verify via PubKey
- Digital Certificate (identify the client and the server)
  - OU Name + Email
  - **Who issued the certificate** (it's signed!)
  - **PubKey** ← (i.e. to retrieve the «official» PubKey of Webserver)
  - X.509 format based on ASN.1 (PEM+DER)

Tools for a security approach (4)

- Certification Authority (organization that issue certificate)
  - Self signed
  - **Root-of-trust** → Who issue digital certificate
    - To enforce check, i.e. Web Browser have a complete of official CA list to validate Web Server PubKeys «for domain name»
- To create Client Certificate
  - Certificate Signing request CSR with
    - Signed with PrivKey of Client (to enforce Identify)
  - CA return a certificate with PubKey of Client (To enforce Identity)
    - + Sign with PrivKey della CA (to enforce Root-of-trust)

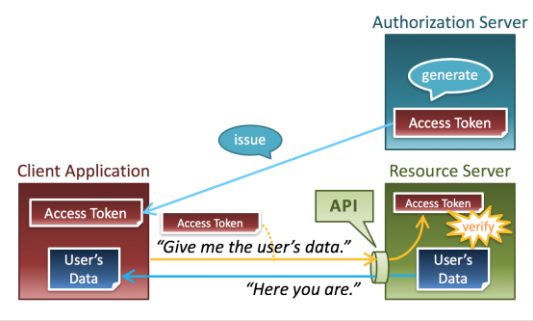
Tools for a security approach (5)

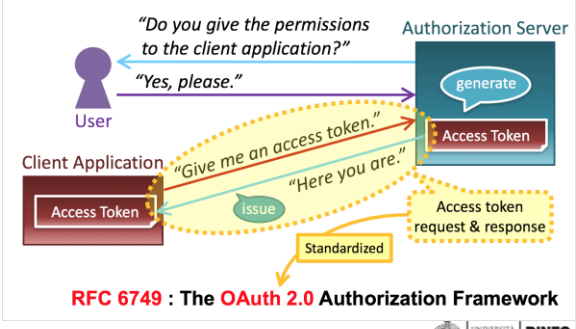
- HTTPS on top of HTTP (always!!!)
  - **Protect (almost) everything** (except IP, Port, length of data) via SSL/TLS
  - Long term **PrivKey/PubKey** → cert X.509 server+client + CA
  - Short term **SESSION-ID** → symmetric for any **connection**



OAuth2 (Authorization)

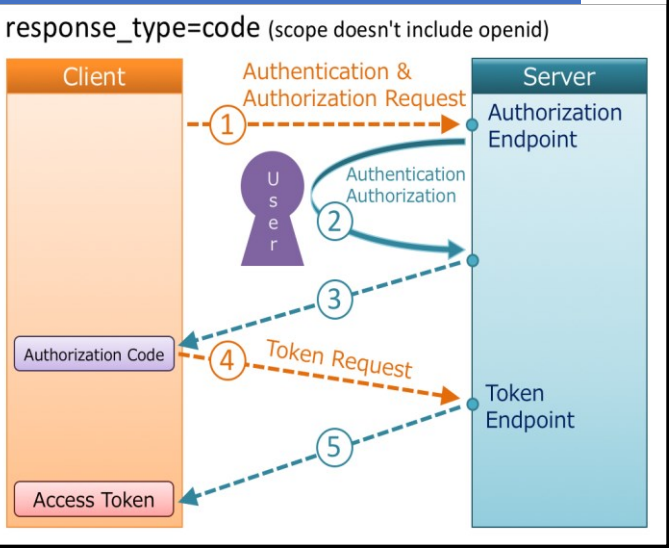
- **Protocol** allows third-party applications to grant limited access to an HTTP service, either on behalf of a resource owner or by allowing the third-party application to obtain access on its own behalf



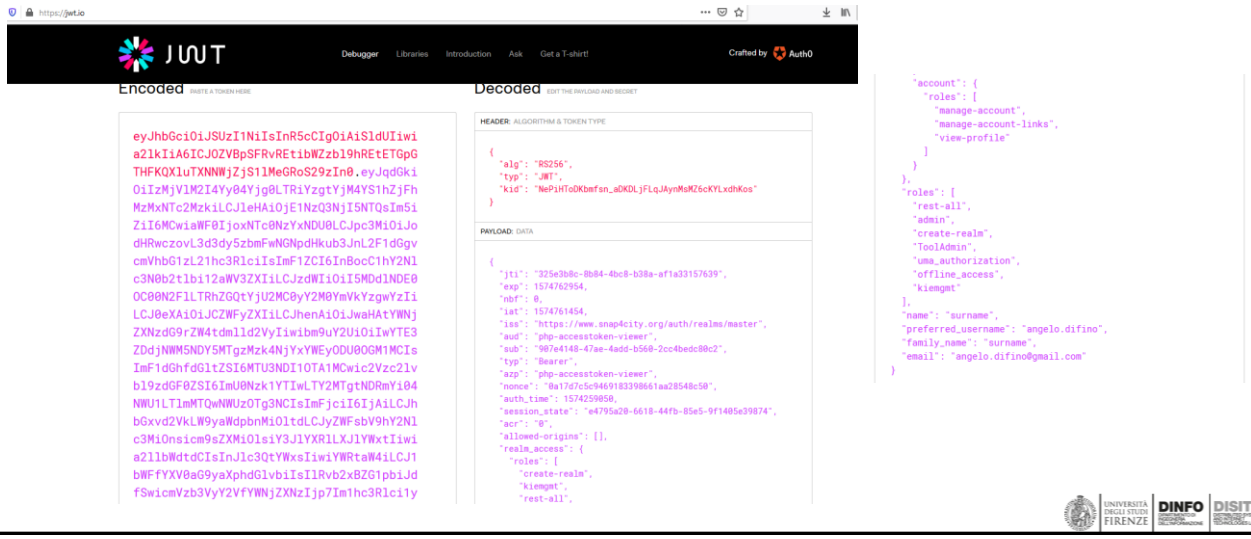


OpenID Connect (Authentication)

- **Identity layer** on top of the OAuth 2.0 protocol, which allows computing clients to verify the *identity* of an end-user based on the authentication performed by an authorization server, as well as to obtain basic profile information about the end-user in an interoperable and REST-like manner



JWT TOKEN



IoT ecosystem

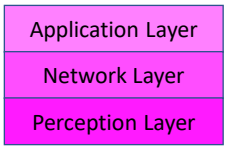
- *"a **dynamic global network infrastructure** with self-configuring capabilities based on standard and interoperable communication protocols where **physical and virtual 'Things'** have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network"* Institute of Network Cultures
- *"a global infrastructure for the information society enabling **advanced services by interconnecting** (physical and virtual) things based on, existing and evolving, interoperable information and communication technology"* ITU-T (2012) Next Generation Networks

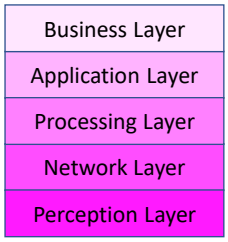
IoT architecture

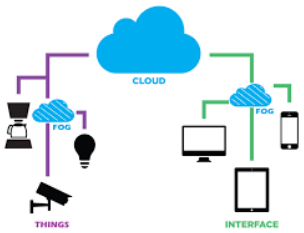
- Independent **IoT ecosystems** that can be
  - physical
  - virtual
  - hybrid mix of the two
- consist of a list of **active** physical devices, sensors, actuators, services, communication protocols and layers, final users, developers and interface layers.

IoT architecture

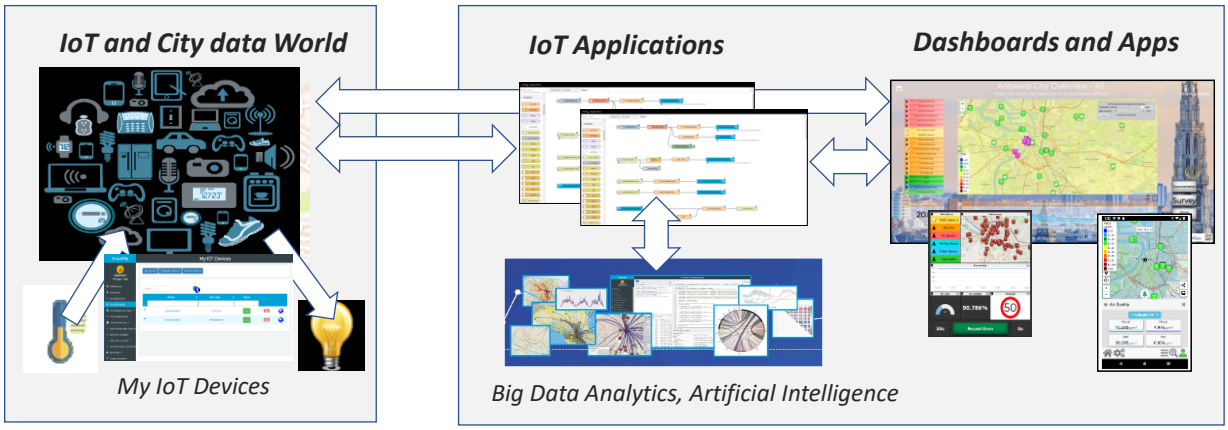
- Several functional blocks are defined in an IoT system, even if a **common conceptualization is not found**, but several different approaches are usual considered: 3-layer, 5-layer, cloud and fog systems, social IoT paradigms.



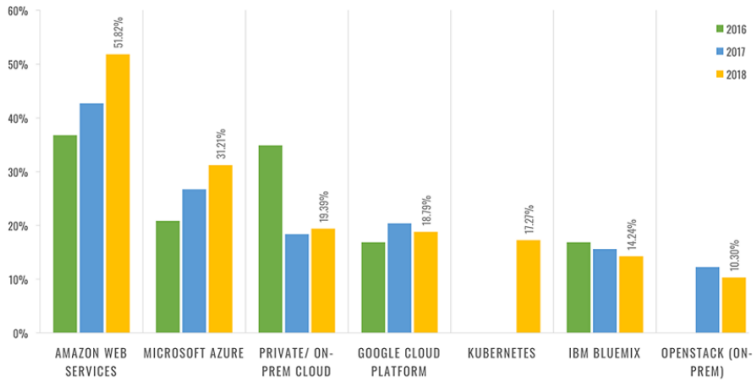




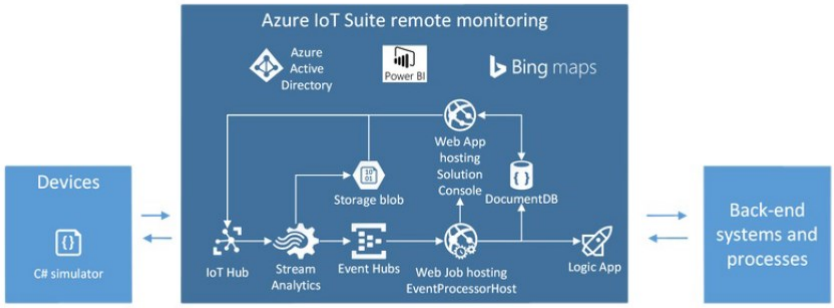
IoT Sentient solutions



State-of-the-art IoT architecture



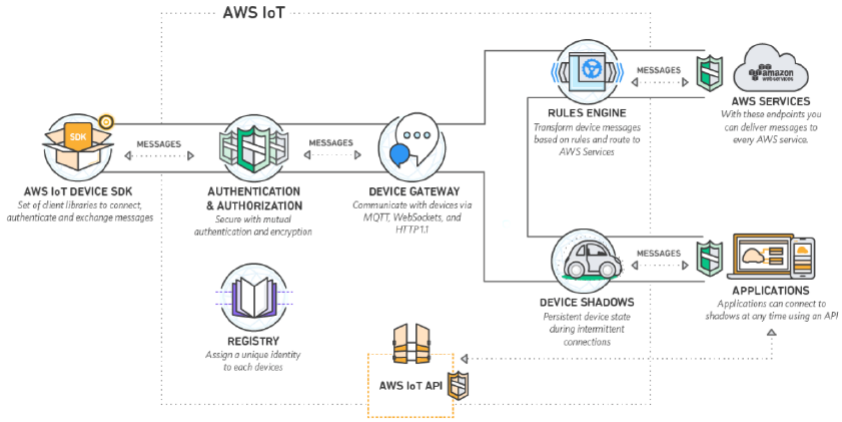
Azure Microsoft IoT (1)



Azure Microsoft IoT (2)

- Hub that communicate with the internal ecosystem
- .NET, Java,Node.js, C, Python
- MQTT, AMQP, MQTT on WebSocket, HTTPS, AMQP on WebSocket
- TLS, SAS Token, IAM, x.509

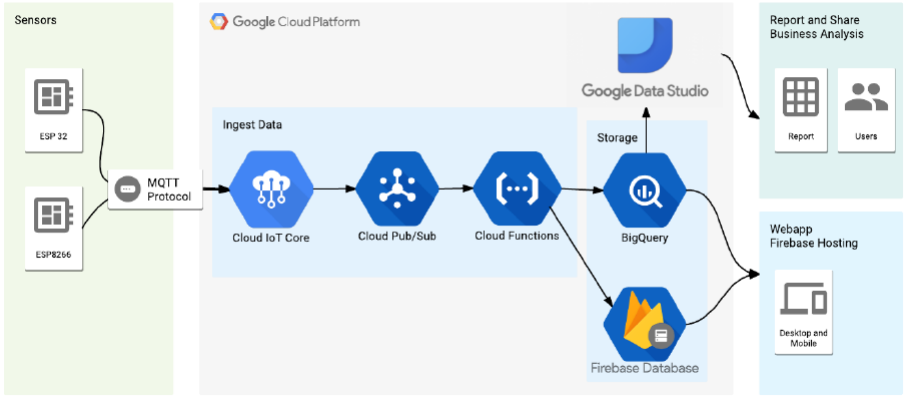
AWS – Amazon IoT (1)



AWS – Amazon IoT (1)

- Data collected by Rules Engine and from the Device Shadows.
- C, Javascript, Java, Python, IOS, Android, Arduino Yun
- MQTT, MQTT on WebSocket, HTTPS
- TLS, x.509, IAM, Amazon Cognito, Federated Identities

Google IoT

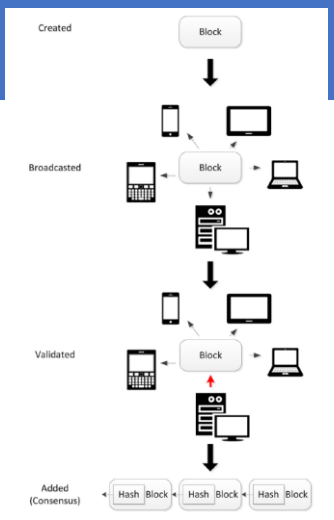


Google IoT

- Core that communicate with internal functionalities, in a Pub/Sub and Dataflow manner
- Go, Java, .NET, Javascript, IOS, Android, PHP, Ruby, Python
- MQTT, HTTP
- JSON Token, IAM, x.509

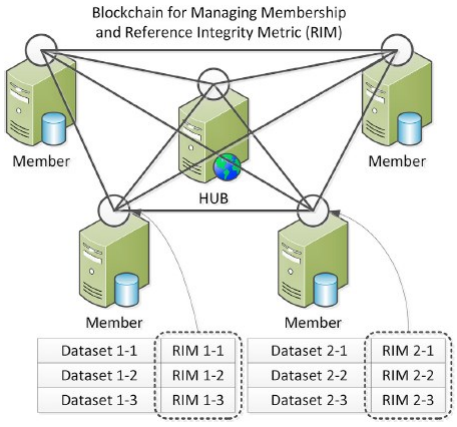
Blockchain solution (1)

- One node validates the block (called mining in bitcoin) and broadcasts it back to the network.
- The nodes add the block to their chain of blocks if the blocks is verified and the block correctly references the previous block

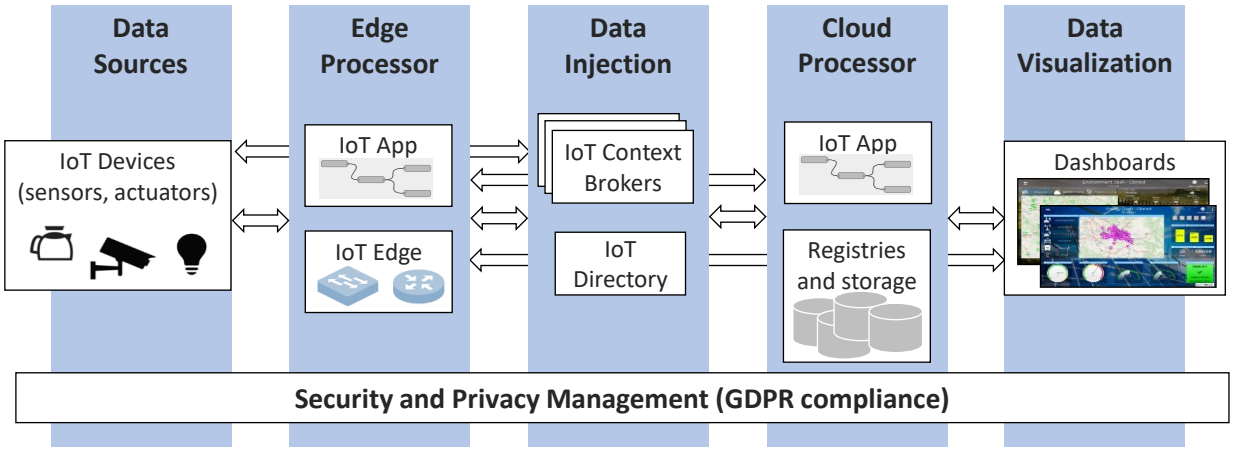


Blockchain solution (2)

- Central hub that maintains references of member repository where the datasets are actually stored and distributed
- Delete from Block chain?
- Rule enforcement (everything distributed)?



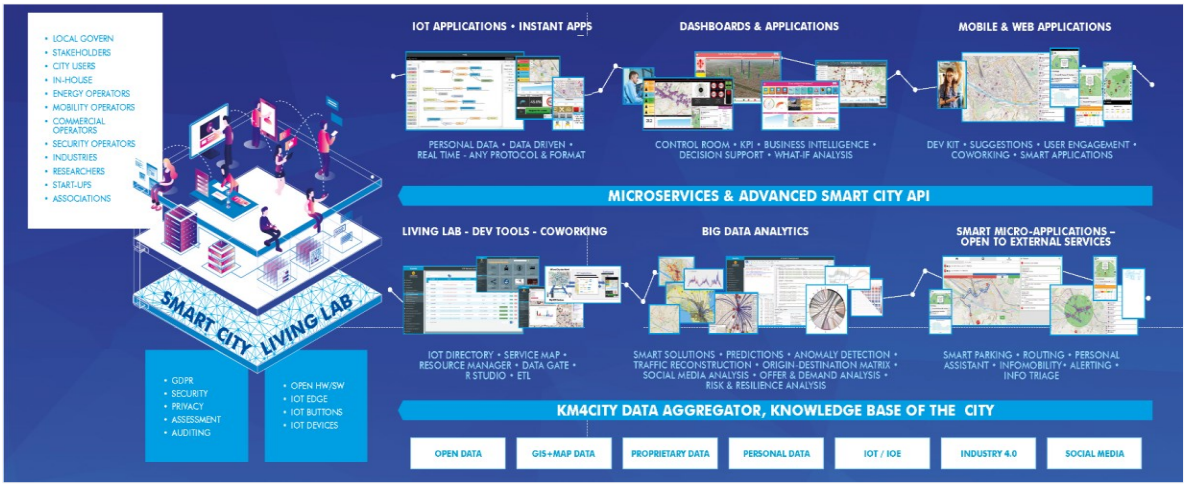
IOT involved entities



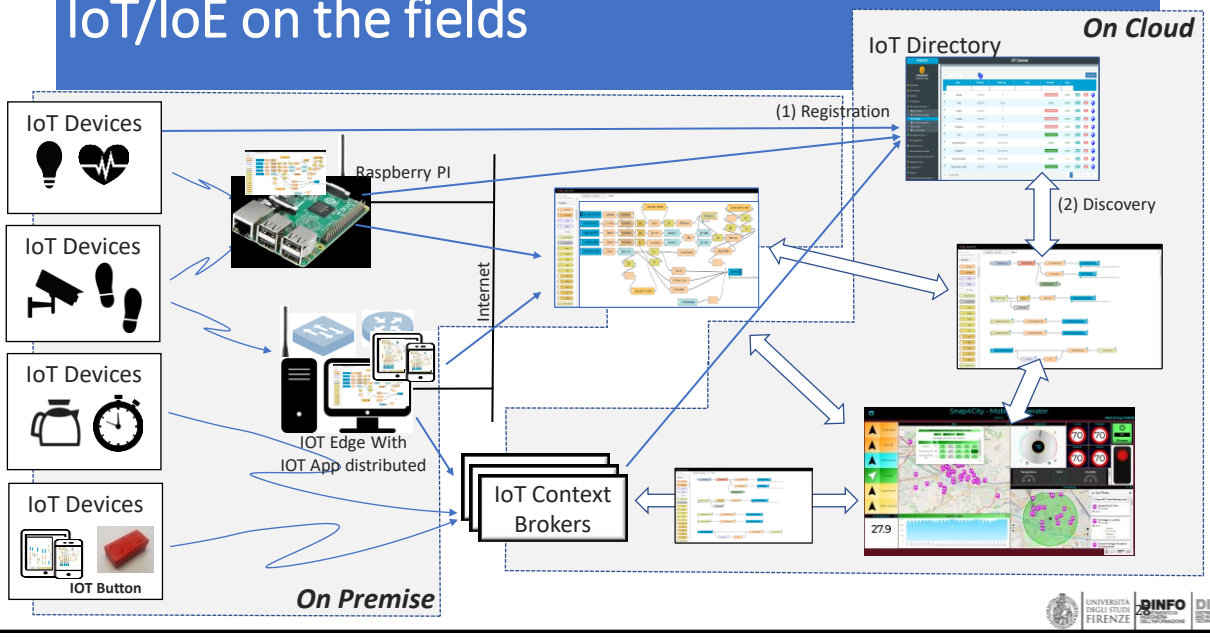
IoT main components

- IoT Device
- IoT Router (with/without computation capabilities)
- IoT Broker (+ Shadowing)
- IoT Device Directory
- IoT User Management
- IoT Service Bus (Pub/Sub, Rule-engine, Data-driven)
- IoT Analytics
- IoT Data repository
- IoT Applications (off-grid/on-cloud)
- IoT Dashboards

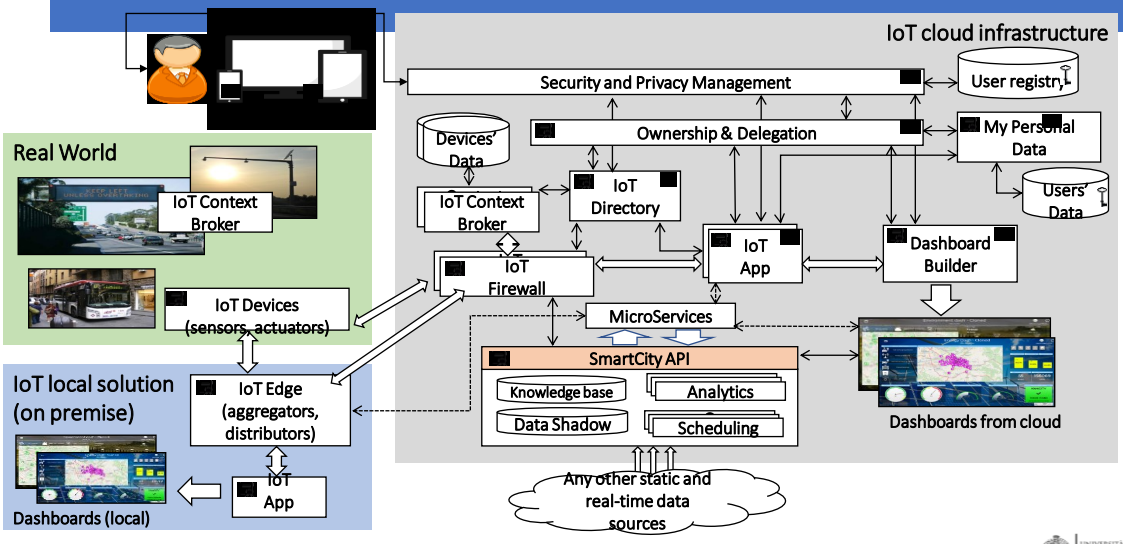
SNAP4CITY platform



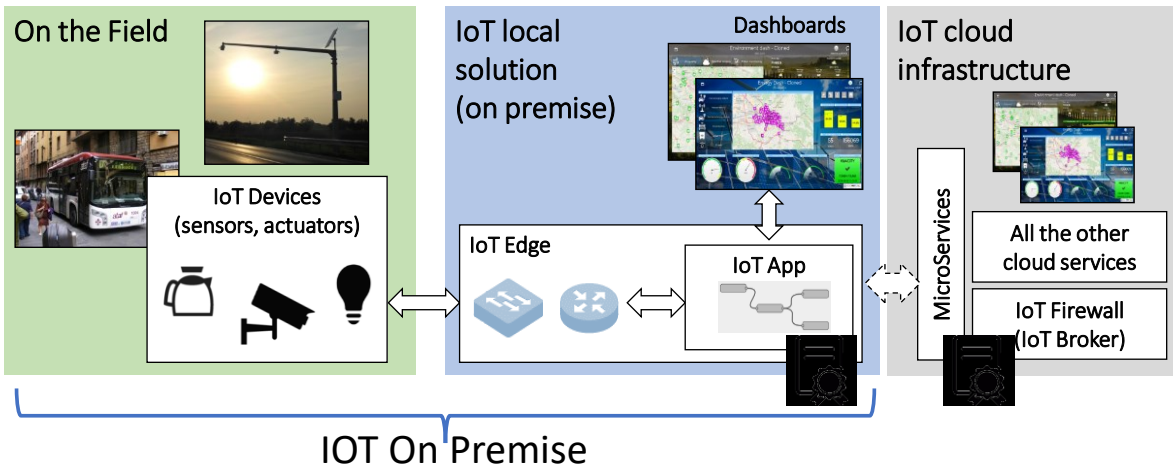
IoT/IoE on the fields



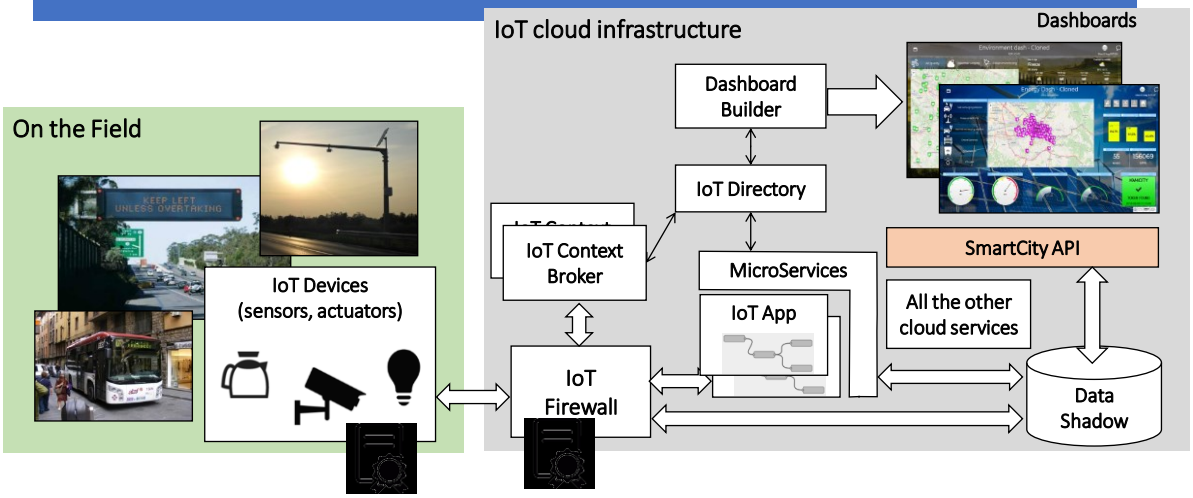
Generic IoT architecture



IOT on premise vs on cloud



IOT Devices vs on Cloud Platform

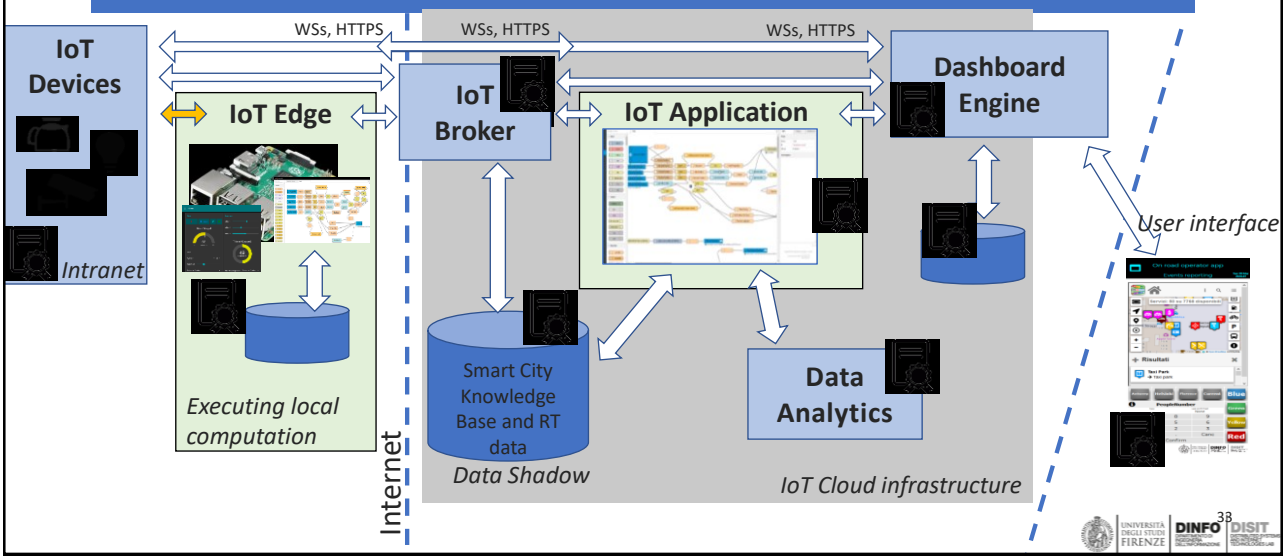


31

Requirements

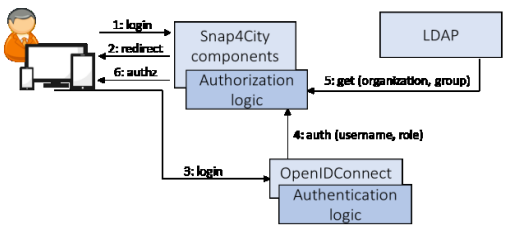
- **Supporting security among**
  - IoT Brokers, IoT Discovery, IoT Applications, Dashboards, Storage, etc…
  - Authenticated Connections: H2M, M2M
  - Secure Communications: H2M, M2M
  - Authorization according to the role, group, organization of the user
- **Deliver Open Software on well known platforms, end-2-end secure IoT stack**
  - Arduino, ESP32, Raspberry Pi, Linux, Windows, Android, etc.
- **GDPR recommendation:**
  - Individuals must provide explicit consent to data collections
  - Right to be forgotten
  - Provide easy access to individuals data
  - Explanation about how automated decision are computed against personal data
  - Disclosure within 72 hours of data breach
  - Data protection by design

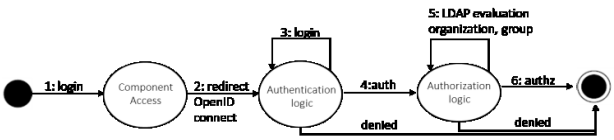
End-to-end security



AUTHENTICATION AND AUTHORIZATION

- **Authentication is performed via OpenIDConnect as (SSo) which is based on OAuth2**
  - User Registry on LDAP/CRM for user data
  - Authenticated users have Role of the LDAP registry
  - Thus Communication start with SSL/TLS protocol, sharing a secret via JWT Token
- **H2M: login is needed**
- **M2M: first time it has to be H2M**
  - then a Refresh Token is retrieved based on the first JWT





SECURITY AND PRIVACY MANAGEMENT

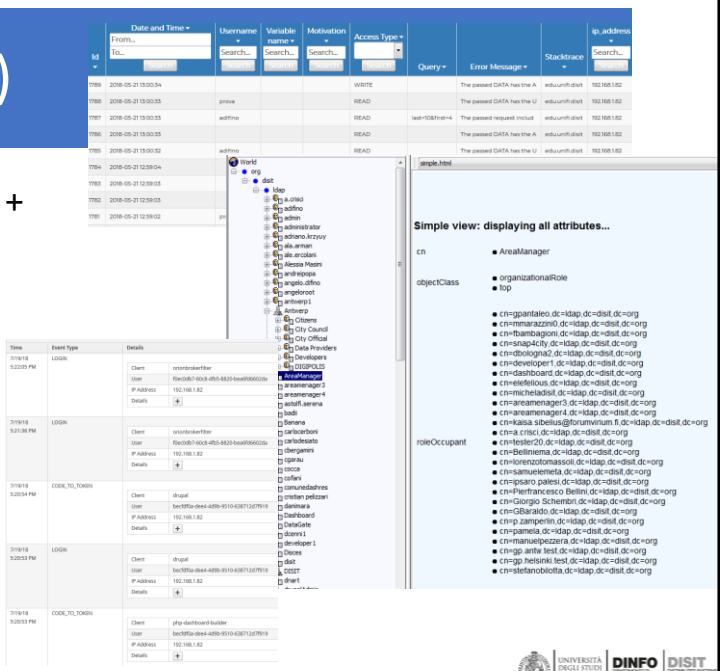
- **From proprietary server:**
  - The device are registered and data collected by the proprietary servers: SigFOX, TheThingsNetwork, etc.
  - SigFOX: the server provides K1, K2 to read the data or subscribe
  - TTN: other kind of keys are used for the same purpose
- **From Open Solutions**
  - K1, K2 can be produced for IoT Device registration, subscription, etc.
  - K1, K2, plus SHA1/3 of Certificate to establish TLS connection
  - Certificate and credentials for the mutual authentications (for TLS connection)
- **Ownership and delegation**
  - Identification of user data type
- **User's group, organization. User's roles**
  - User's grants and rights to access data
- **Auditing, right to be forgotten**
  - Values, Devices, Brokers, IoT App, Dashboards, User Profiles, time series, etc.
  - Data breach intrusion detection
- **Assessment**
  - User and device limit constrains

On regards GDPR (1)

- Assessment and auditing
- CMS for personal data information, encryption
- Explicit Consent, Ownership and delegation
- Roles and organization (groups) to permits fine access control
- Any collected data labelled with
  - Data of collection
  - Data of injection
  - Data of elapsing
  - Data of deleting
- +process to purge elapsed data

On regards GDPR (2)

- Unified Login → via Keyclock + LDAP
- My Personal Data
- Data auditing
- Federated modules
- IoT Directory and certificates
- IoT Button
- IoT Dashboard



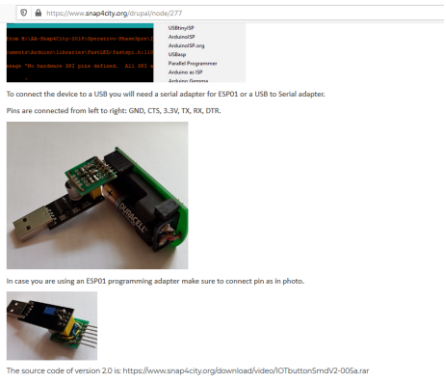
Any Devices in the IoT ecosystem

- Microcontroller ESP8266
- Microcontroller Arduino
- Raspberry boards
- Android devices
- PC
- On cloud virtualization

Security Strength	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80	2TDEA[21]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160$-223
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224$-255
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256$-383
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384$-511
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

- As much as user friendly VS as much as secure channel
- On embedded devices, cypher suite not always available. Use: TLS_RSA_WITH_AES_256_CBC_SHA
- Impact of certificate size on available heap: NIST Special Publication suggestions: Use 2048, but WARNING!

Any Devices in the IoT ecosystem (2)



Any Devices in the IoT ecosystem (3)

More on breach (1)

- Dangerous examples of network vulnerabilities include
  - Improperly configured routing causing leak paths in between protected network enclaves or to the Internet itself
  - Temporary or test configurations of firewalls that don't operate as designed or don't get reversed-out properly
  - Password password password
  - Password password password
  - Network analysis in real-time → dashboards, acceptable level of traffic, trigger of alarm → Notification (SMS, Mail, Calls leveraged depending on the sensitivity)
- Two authenticate factors → FIDO2 with hardware support (SOLO)

More on breach (2)