Data management for early warning & DSS

Sensors' data collection, organization, knowledge extraction and redistribution

SENSORIUM

Computational Modelling Laboratory

In a nutshell

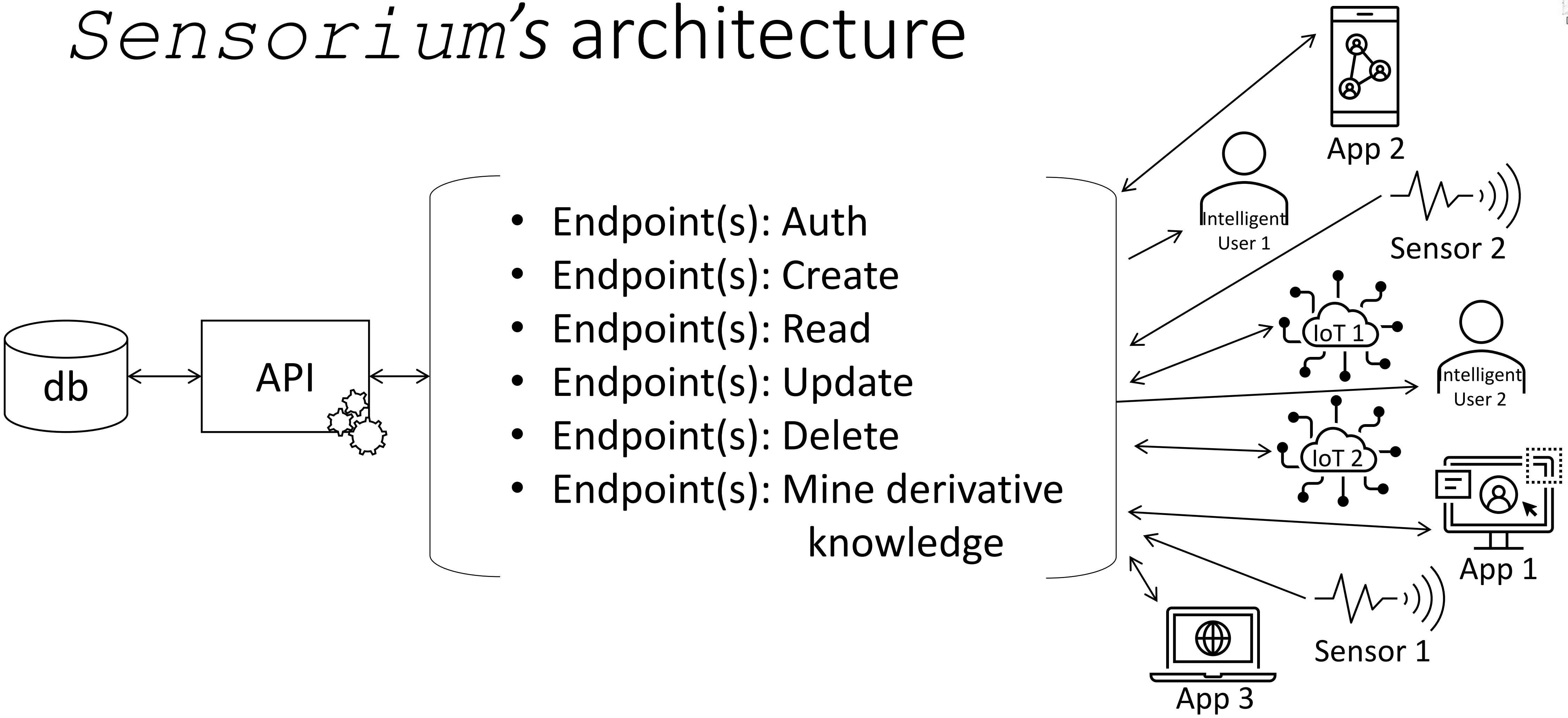
- CMODLab creates, collects, curates and redistributes an impressively large volume of information related to its fields of interest, especially fires
- A permanent, scalable, secure, modern, accessible and efficient infrastructure is required to manage this information
- A database system and RESTful API for CRUD & knowledge mining operations on the aforementioned information are required
- This is `sensorium` (the entire system), and `salvaroba` (the database).

Computational Modelling Laboratory

Field information management areas

- Civil protection
  - Project Telemachus: earthquakes
  - Project Laertes: landslides
  - Project Aetheras: water quality, water pollution, coastal erosion & coastal microclimates, marine alien species
  - Hazards' assessment of geological & anthropogenic hazards in the Ionian Sea
- Primary sector
  - Project DAKOS: olive cultivation
  - Project e-Olive: olive cultivation
  - Project Olive Watcher: olive cultivation
- Biodiversity – Environment
  - Project BIG: NATURA 2000 and endangered species
  - Project BIONIAN: biodiversity in the Ionian Sea
  - Project BEST
- Smart Cities
  - Project PeopleFlows: Measuring people in public
- Water resources
  - Project SaveWater
- Primary sector
  - Microclimates
- Systems simulation and multimedia and augmented reality information management
  - In IoMT networks
  - For systems co-creation
  - For cultural information

Sensorium's architecture



Computational Modelling Laboratory

Selecting a hybrid DB (1/2)

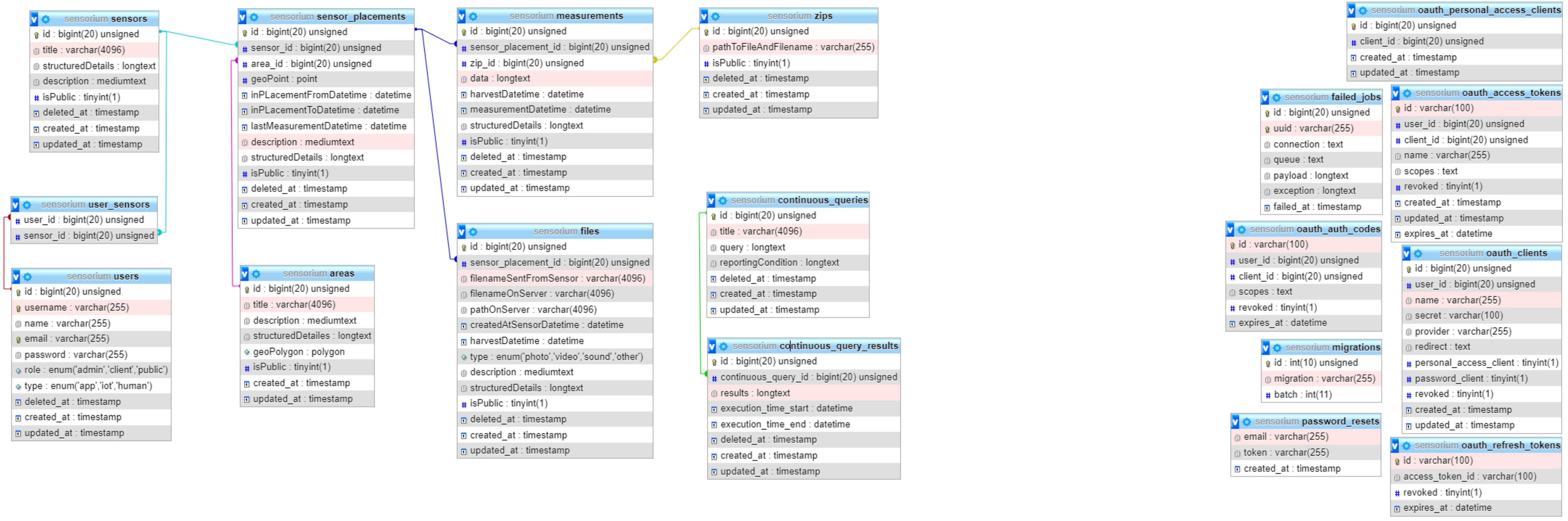
- Best of both worlds
  - Keep stable event envelope relational
    - event_id, locaton_id, device_id, event_time, status
  - Store variable event payload in structured json / jsonb
  - MySQL / MariaDB / PostgreSQL explicitly supports JSON querying & JSON indexing
- Integrity where it matters
  - Use relational constraints for the parts that must stay correct
    - keys, uniqueness, and links to canonical entities like grouping notions, devices, and accounts.
  - JSON lives alongside normal SQL data with transaction support
- Better fit for mixed workloads
  - Streaming platforms rarely do ingestion only
  - Need joins to metadata, deduplication, rollups, “latest per device,” rankings, and time-windowed analysis = native SQL processes

Selecting a hybrid DB (2/2)

- Flexible schema evolution without chaos
  - New event attributes -> JSON without forcing constant table redesign
  - Later, fields identified operationally important = indexed
- More disciplined scaling for large streams
  - Relational systems support declarative partitioning
    - strong fit for append-heavy event data partitioned by time & other routing keys
- Cleaner downstream model
  - Hybrid design =
    - core business entities normalized & governed
    - “long tail” / irregular event attributes stay flexible
  - Reduces duplication + makes reporting, billing, and compliance-style queries easier

Database *salvaroba architecture*

Computational Modelling Laboratory



General description of `sensorium`

- It’s a generalized repository for sensor information and measurements
- It’s agnostic as to what kind of sensors it hold the information of
- It’s a Database and an interface for programmatic access (API)
- It has a set of procedures that, despite being agnostic about the data, mine new, non-obvious, difficult to extract manually, and useful (actionable) information, both based on user request and based on the data (continuous querying)

Sensorium's users (1/2)

- They are divided into categories
  - Applications (extensive programs- applications with their own business logic, e.g. SaaS Olive Observer)
  - Sensors (individual information creating / sending systems, e.g. a smart trap)
  - Io(M)T (interactive information creating / sending - receiving agent systems, e.g. a spraying drone)
  - Living / intelligent users (non-programs, e.g. a human administrator)
- Applications
  - Each maintains a unique user account in the sensorium
  - They manage their user accounts internally (in another DB)
  - They use the sensorium as a repository for their sensor information, while using another DB for their business logic
  - They manage their sensors in the sensorium on their own initiative

Sensorium's users (2/2)

- Sensors
  - They maintain a user account in sensorium
  - After registering in sensorium by the administrator (or an application) they communicate directly with sensorium
- Io(M)T
  - Each such device maintains a unique user account in sensorium
  - They can send information (function as a sensor)
  - They can request information from the sensors to which they have access (e.g. to draw conclusions and act on it)
- Intelligent users
  - The API interface is not intended for humans, still access is allowed for specialized users (e.g. administrator) - also through the Graph API Explorer (future extension)
  - Each such user maintains a unique user account in sensorium

Computational Modelling Laboratory

Sensors' agnostic handling (1/2)

- For `sensorium`, everything that creates information is a sensor
  - drone with a camera,
  - submarine with sonar,
  - arduino for temperature measurement,
  - people filling out a questionnaire,
  - program / application that accepts or creates information, etc.
- All it requires is
  - a consistent method of recording information (generally)
  - (manual correction in case of errors in consistency)

Sensors’ agnostic handling (2/2)

Hybrid database

- Document model: JSON-type fields
  - “**`structured_details`**”: field according to the structuring protocol
  - It is up to the sensors what information they will send – it is not predefined
  - It is searchable, participates in query conditions
- Relational: Explicit structuring
  - “**`title`**” (string)
  - Explicitly predefined structuring of properties (the expected / basic ones) per entity

Supported measurement types

Each measurement can be specified by

- The number of dimensions, in
  - single-measurement: one phenomenon is measured
    - e.g. measurement of humidity only
  - multi-measurement: more than one phenomenon is measured
    - e.g. simultaneous measurement of humidity and temperature
- The number of aspects / manifestations of the phenomenon
  - univariate: from the measurement of the phenomenon a single value / sample is obtained
    - e.g. humidity: 32%
  - multivariate: from the measurement of the phenomenon more than one value / samples are obtained
    - e.g. geographical location: lat:39.6235866, lng:19.9236021

Supported dispatch types

2 types of data sending from sensors

- Sending a single measurement
  - For infrequent measurements
  - For measurements where it is important to take action on time
- Aggregate sending of more than one measurements
  - For high sampling rate
  - For measurements where it is not important to take action on time
  - To reduce sending costs
    - Financial: for networks with volume-charging (e.g. GSM)
    - Sensor’s energy: for IoT with high energy consumption during sending action

Continuous querying infrastructure (1/4)

Capability for continuous queries

- Allows the construction of a query of continuous validity
  - E.g. Find all measurements that are within 5% of the maximum
  - E.g. Find the measurements that spatially belong to a polygon
- These are executed automatically after each addition / change of data
  - They do not require action from the administrator
- They can automatically inform users or the database of their results
- Value proposition
  - From raw measurements to actionable events
  - From passive storage to active monitoring
  - From manual querying to always-on intelligence
  - From single-sensor readings to system-level situational awareness

Continuous querying infrastructure (2/4)

- **Threshold breach detection**
  - Detect values above / below critical limits for temperature, humidity, vibration, gas concentration, water quality, etc.
- **Rate-of-change alerts**
  - Detect sudden increases or decreases, not only absolute values
  - Example: temperature rise of more than X% within Y minutes
- **Trend persistence detection**
  - Trigger only when condition remains true for a min duration
  - Reduces noise from short-lived spikes
- **Multi-sensor confirmation rules**
  - Require agreement from 2+ nearby sensors before raising an alert
  - Improves reliability and lowers false positives
- **Spatial containment / proximity checks**
  - Detect measurements inside protected zones, near coastlines, near settlements, or within distance from critical infrastructure
- **Moving object monitoring**
  - Track whether drones, vehicles, vessels, or mobile sensors enter forbidden or high-risk areas
- **Cross-variable correlation queries**
  - Detect combinations that are weak individually but important together
  - Example: high temperature + low humidity + strong wind
- **Baseline deviation detection**
  - Compare current values against normal values for that location, season, or hour of day
  - More meaningful than static thresholds
- **Anomaly / outlier discovery**
  - Detect measurements that differ strongly from neighboring sensors or from historical behavior

Continuous querying infrastructure (3/4)

- **Sensor health monitoring**
  - Identify frozen sensors, missing transmissions, abnormal repetition, impossible values, or battery/network degradation
- **Data quality control**
  - Flag inconsistent units, malformed payloads, duplicates, timestamp anomalies, and spatially impossible readings
- **Continuous ranking queries**
  - Maintain “top-N” lists in real time
  - Example: hottest locations, most active traps, most polluted stations
- **Window-based event aggregation**
  - Continuously summarize the last 5 minutes / 1 hour / 24 hours
  - Example: averages, maxima, counts, exceedance frequency
- **Early warning escalation logic**
  - Convert raw detections into alert levels
  - Example: advisory → warning → emergency based on severity and persistence
- **Derived knowledge generation**
  - Automatically compute risk scores, stress indices, composite indicators, or hazard maps from raw observations
- **Rule-driven notification routing**
  - Send different alerts to different user categories
  - Administrators, applications, field operators, citizens, Io(M)T devices
- **Automatic actuation support**
  - Feed query results directly to decision-support or automated response systems
  - Example: irrigation trigger, drone dispatch, inspection request
- **Project-specific continuous intelligence**
- **Same infrastructure, different rules per domain**
  - Fire risk, landslides, water pollution, olive cultivation, biodiversity, smart cities

Continuous querying infrastructure (4/4)

Is it **actually** possible in real-life cases?

- Yes: most continuous-query ideas are compatible with hybrid model
- Best-fit cases
  - thresholds, joins to metadata, rankings, rolling aggregates, spatial containment, baselines, data quality
- Some care required for
  - rate-of-change, persistence, alert escalation, sensor health
- Few stay outside DB core
  - rich anomaly detection, external notifications, device actuation
- Key pattern
  - relational event envelope + JSON payload + generated/indexed JSON paths + spatial columns + triggers/events for automation

Mining knowledge (1/3)

Sensors’ value prediction with measured certainty

- Move from observation to anticipation
- Support early warning before thresholds are crossed
- Turn raw time series into probable future states
- Help users act earlier, not only react later
- Make predictions usable by attaching certainty / uncertainty

Mining knowledge (2/3)

What the system should predict

- Next value / next state
- Value at a future horizon
- Probability of threshold exceedance
- Expected time until critical event
- Expected trend direction and speed
- Best / expected / worst plausible scenario

Why “measured certainty” matters

- A prediction without certainty is hard to trust
- Users need risk, not only point estimates
- Decision-making improves when confidence is explicit
- Certainty helps distinguish weak signals from strong warnings
- The system can escalate only when confidence is sufficient

Mining knowledge (2/3)

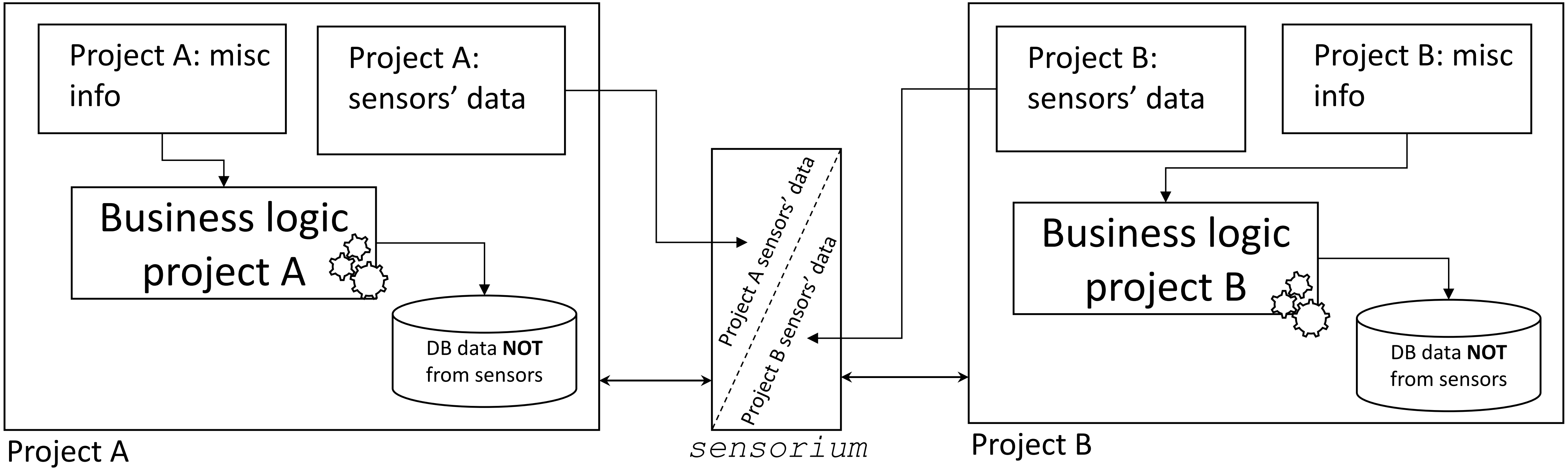
Generic classes of prediction methods (under dev)

- Baseline / naive forecasting
  - Moving average / rolling trend
- Classical statistical time-series methods
  - Trend + seasonality models
- Filtering / state-estimation methods
  - Bayesian filtering
- Regression-based prediction
  - Tree-based regression
- Spatiotemporal methods
  - Neighborhood-based prediction
- Machine-learning forecasting
  - Random forests / gradient boosting
- Sequential / deep-learning methods
  - Recurrent sequence models
  - Temporal convolution models
- Hybrid / knowledge-guided methods
  - Rule-based + learned
  - Domain-model + statistical correction

Support for multiple projects

`sensorium` supports all our (**currently foreseeable**) projects

- Each project that works with sensors and their data
  - Gets an account on `sensorium` for its sensors' data
  - Includes its own business logic wherever / however

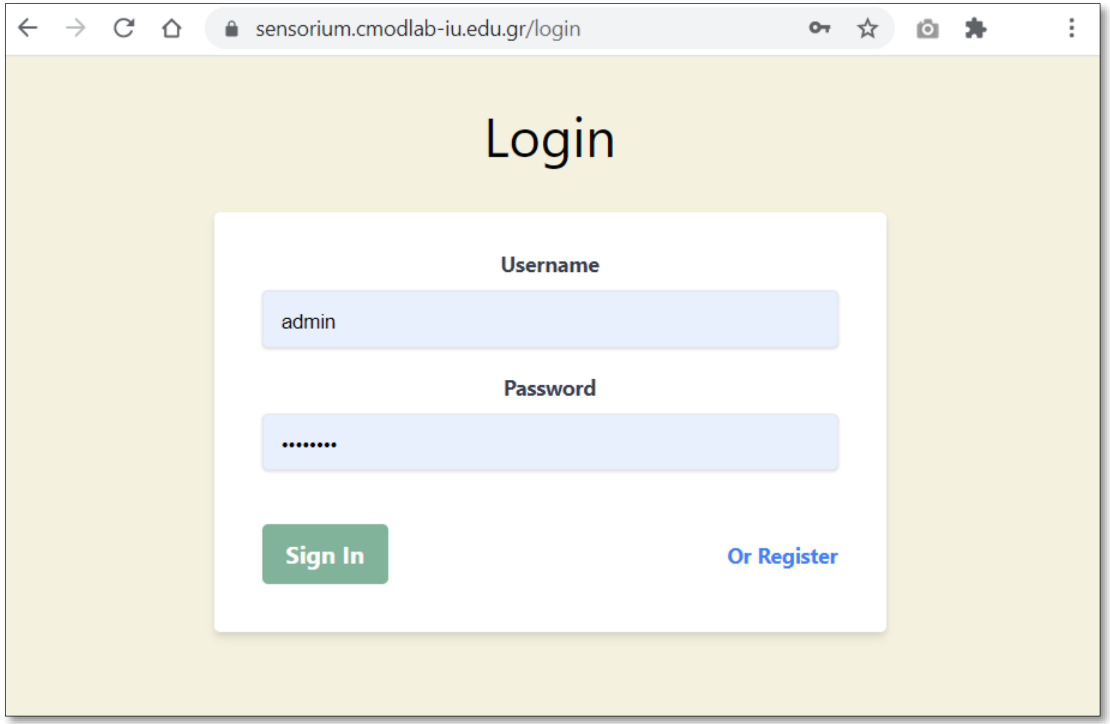


Administrator

Whole of `sensorium`, can execute

- CRUD on users' accounts
- Queries & knowledge extraction

using GUI

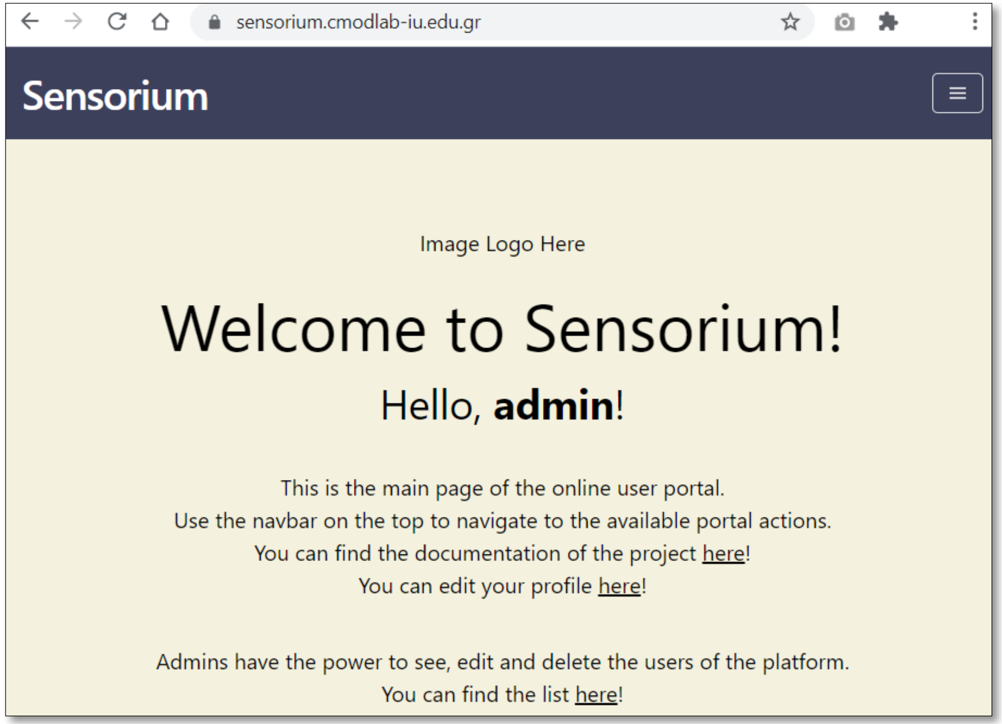


Administrator

Whole of `sensorium`, can execute

- CRUD on users' accounts
- Queries & knowledge extraction

using GUI

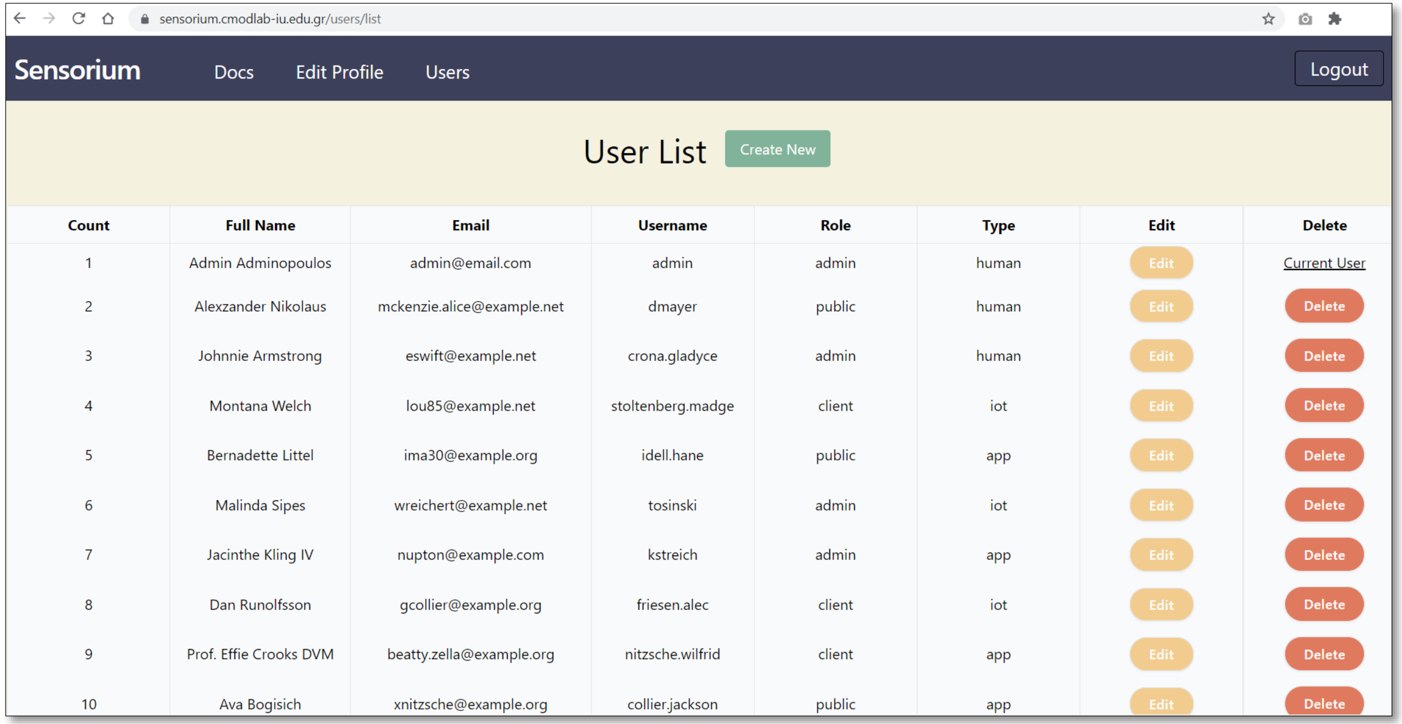


Administrator

Whole of *sensorium*, can execute

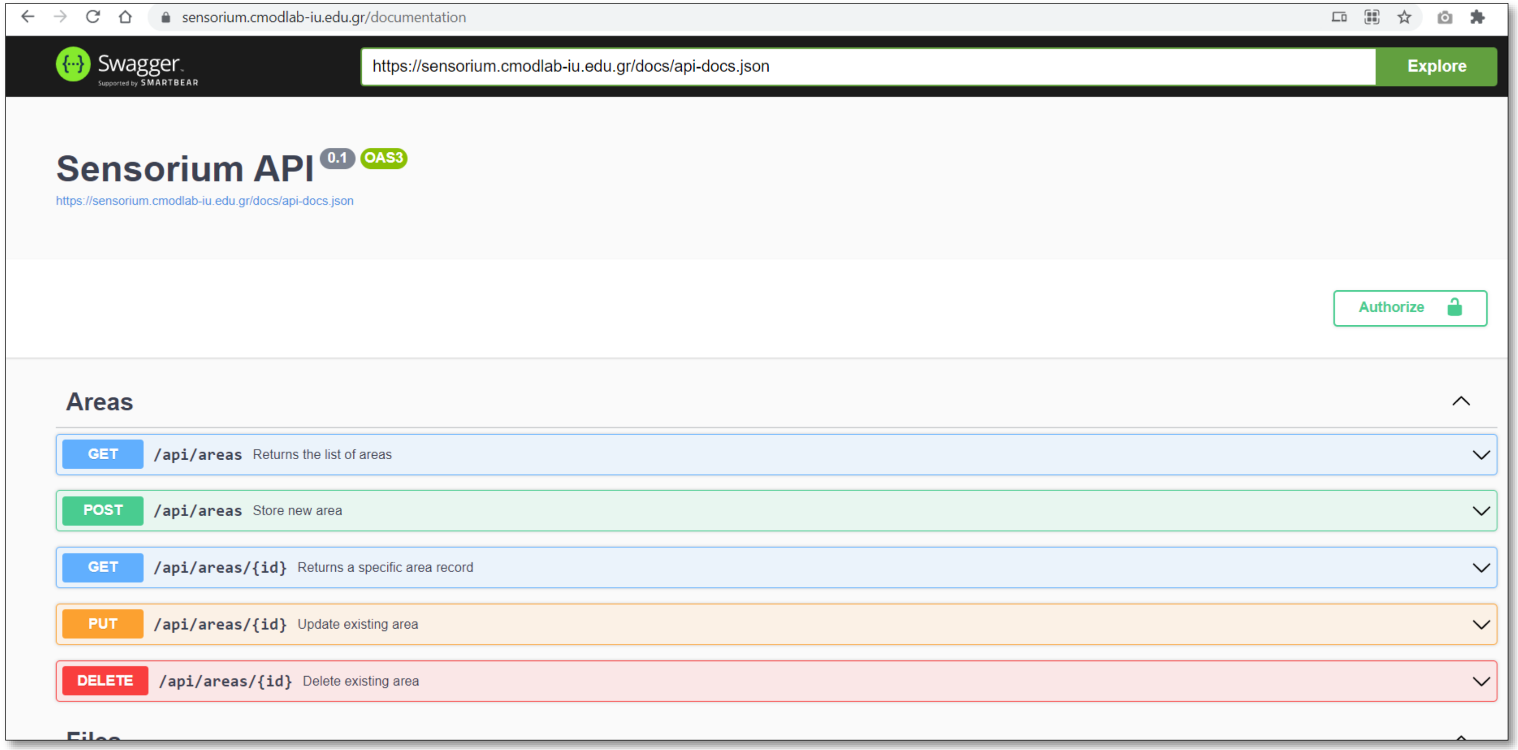
- CRUD on users' accounts
- Queries & knowledge extraction

using GUI



Documentation

- Extensive technical documentation for the API,
- Documentation for
  - the overall system
  - the individual roles

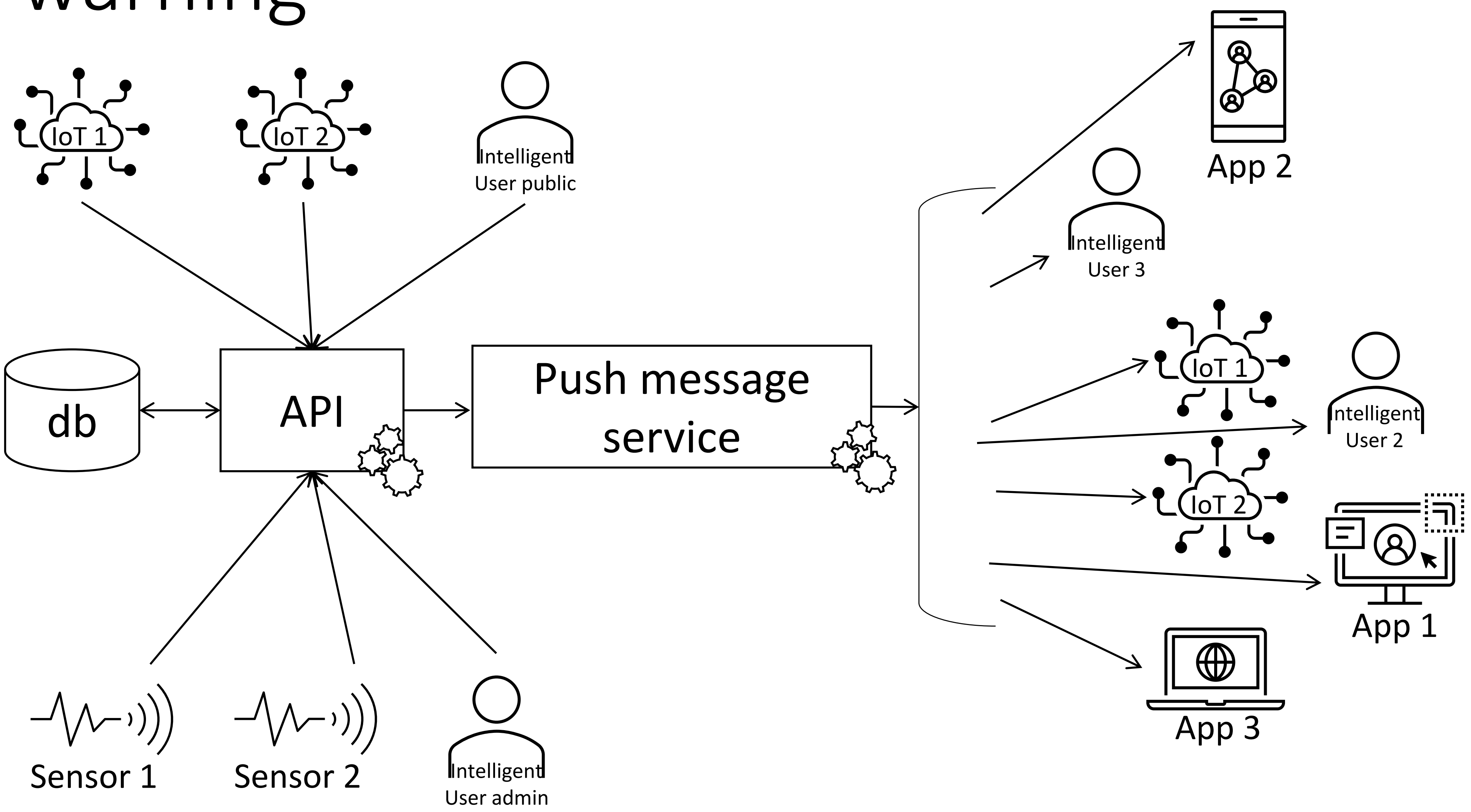


Computational Modelling Laboratory

Planned extensions

- Public API
  - Third-party applications (with / without authentication)
- Graphical interface
  - For public users
  - For authenticated users (except the administrator)
- Programming libraries
  - At least for popular languages
  - For the API: for the convenience of the sensorium-using OS designers
  - For the sensors: for the convenience of the sensor designers
- Backup action organization

Early warning



Scalability (1/3)

- Local installations have limits
  - **But**, projects’ number + flows increase
- Managed cloud MySQL
  - Lowest-friction way to buy more scalability
  - No need for new architecture.
- First escalate operations **then** escalate architecture only if necessary
- Key arguments
  - Pay for scalability when needed, not from day one
  - Faster response to growth spikes

Computational Modelling Laboratory

Scalability (2/3)

Scalability path

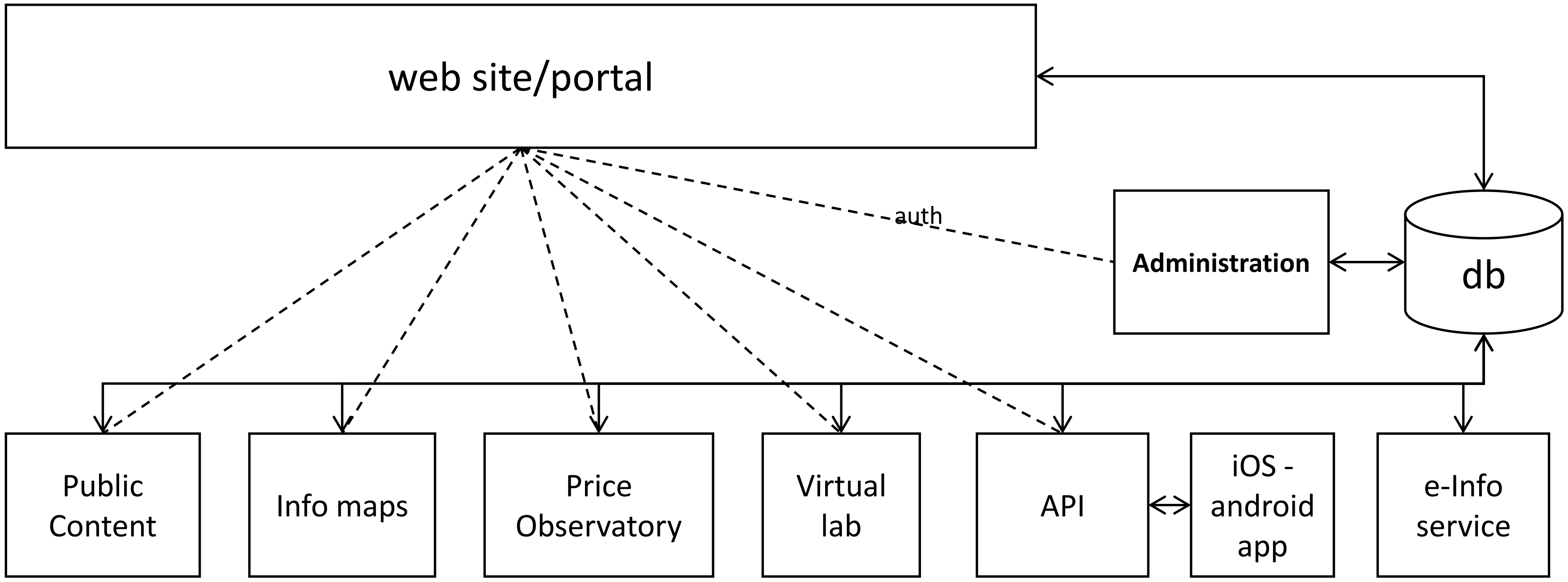
- Keep the data model, change the hosting model
  - Keep schema + move scalability problem into managed platform
  - E.g. Google Cloud SQL & Amazon RDS
- Minimal engineering involvement at the start
  - Migrate database to managed service = provider handles backups, replication, patching, encryption, and storage growth
- Scale up without redesign
  - Straightforward vertical scaling path
    - resize instance -> get more CPU & memory when workload grows
- Scale reads almost as a black box
  - When read-heavy -> add read replicas

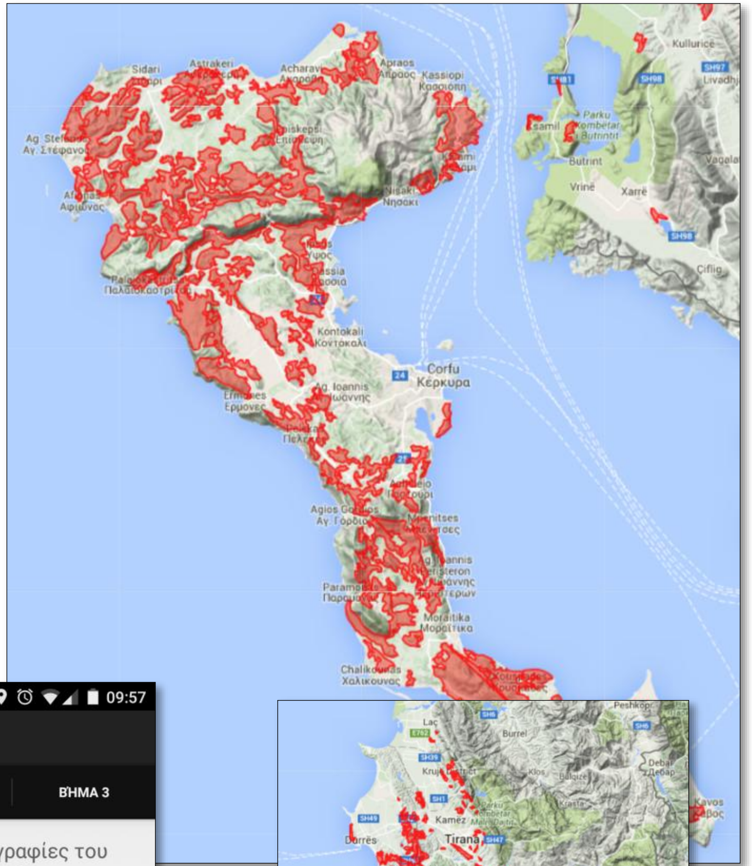
Scalability (3/3)

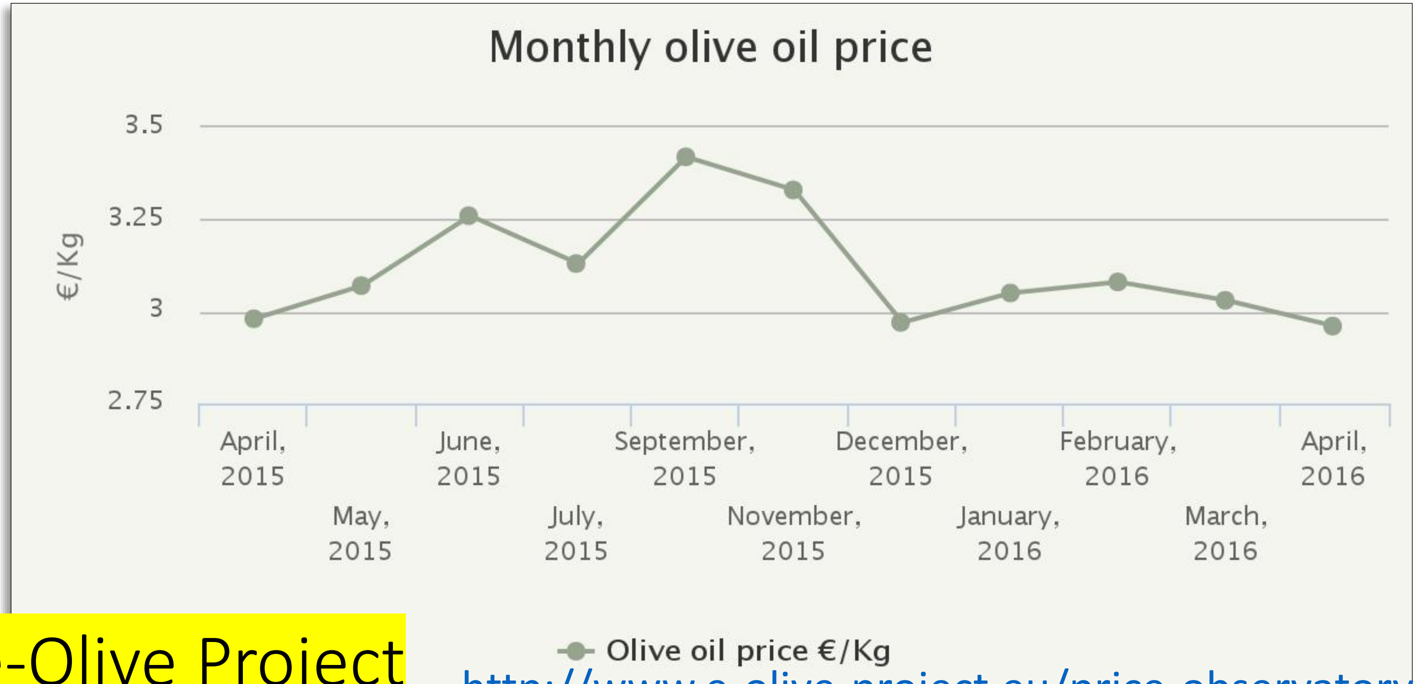
Scalability path

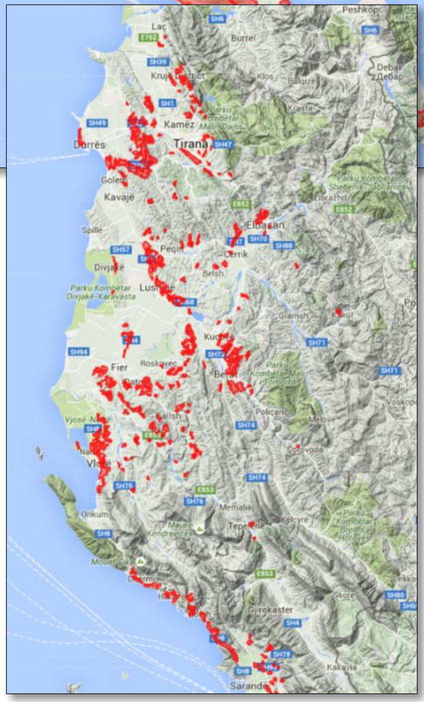
- Storage growth becomes operationally automatic
  - “we are running out of disk” = providers storage autoscaling automatically increases storage when needed
- High availability becomes a managed checkbox, not a custom project
  - Failover & redundancy = product features
- Operational burden shifts from engineers to platform
  - Not only scaling, **but also outsourcing toil**
    - Platform manages backups, recovery configuration, replica management, patch cycles, & misc operations
- Postpone expensive re-architecture (if needed at all)
  - Moving to managed services = headroom before deciding re-design
  - Lower-risk intermediate step

Some of the AgriFood projects we’ve used *sensorium*





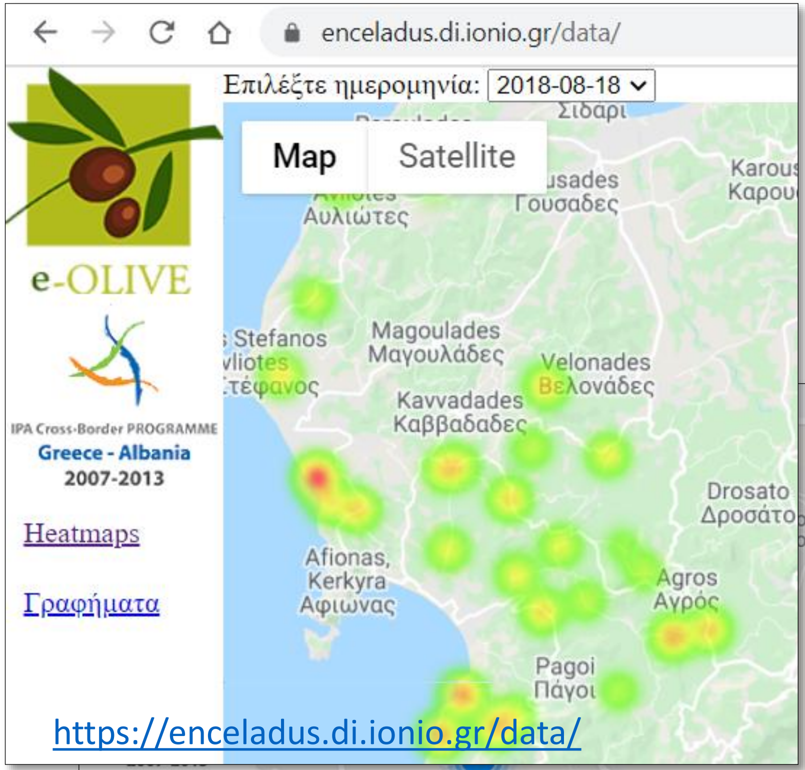




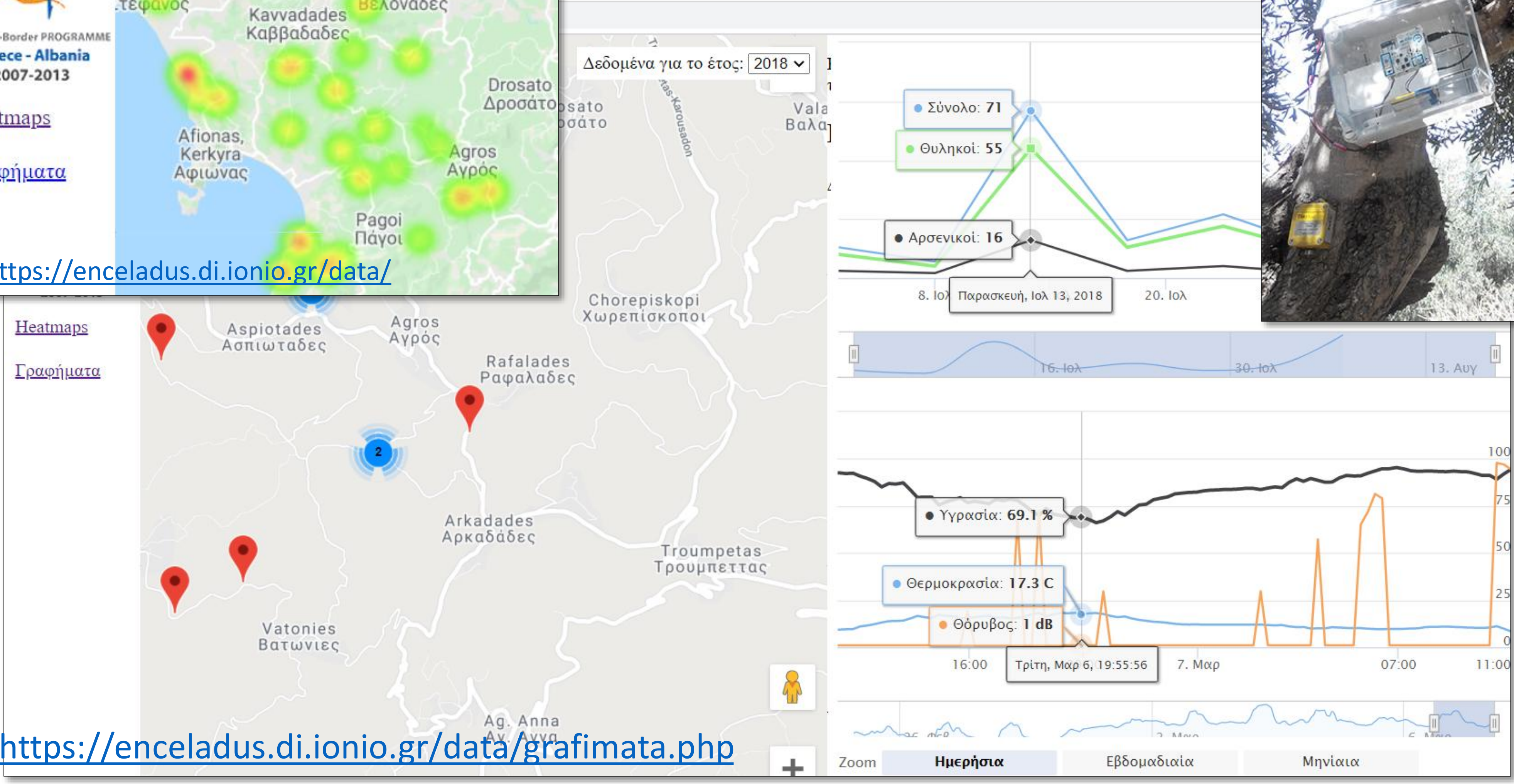
http://www.e-olive-project.eu/xylella-fastidiosa-outbreak/

e-Olive Project

http://www.e-olive-project.eu/price-observatory/

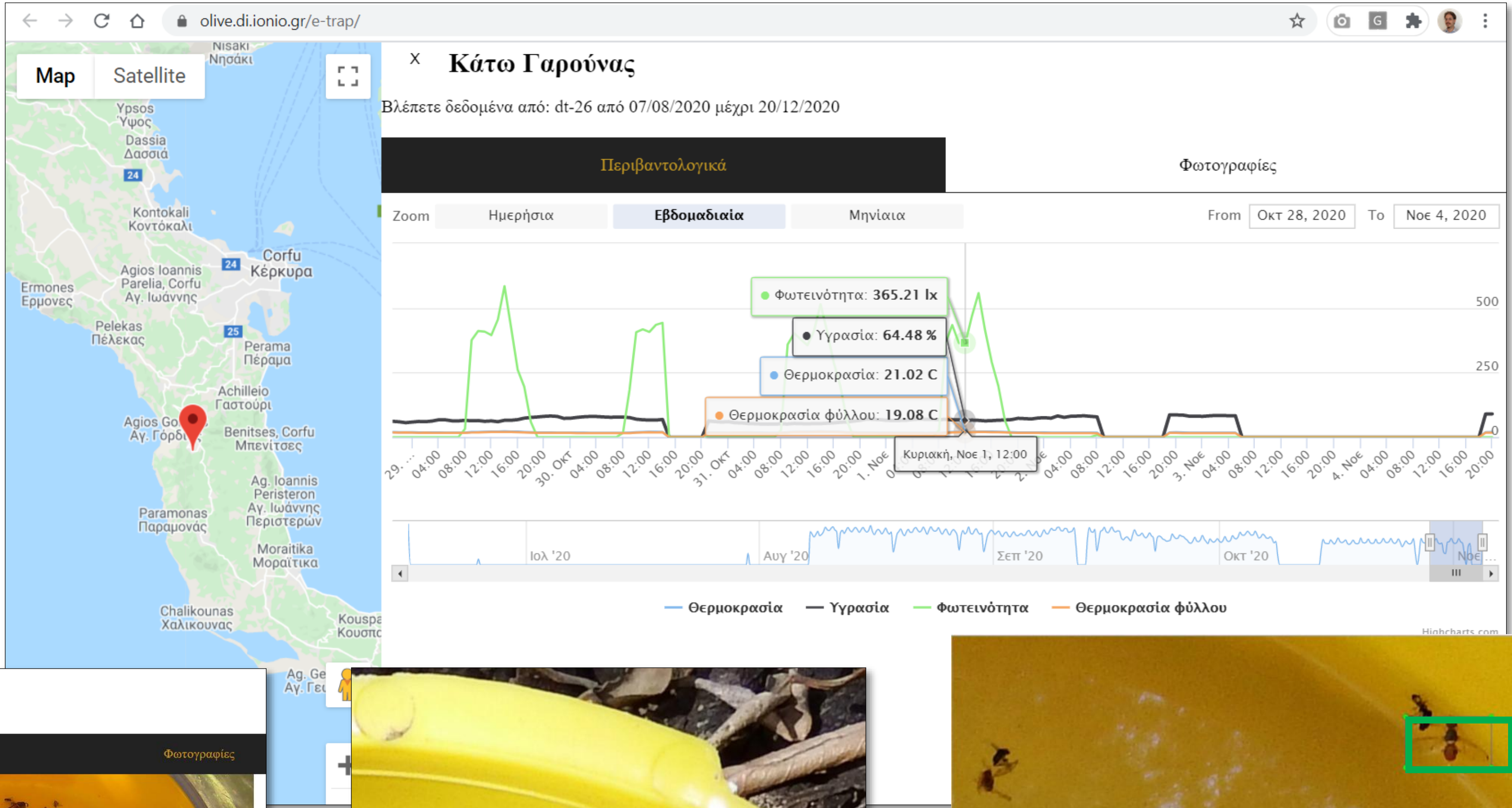


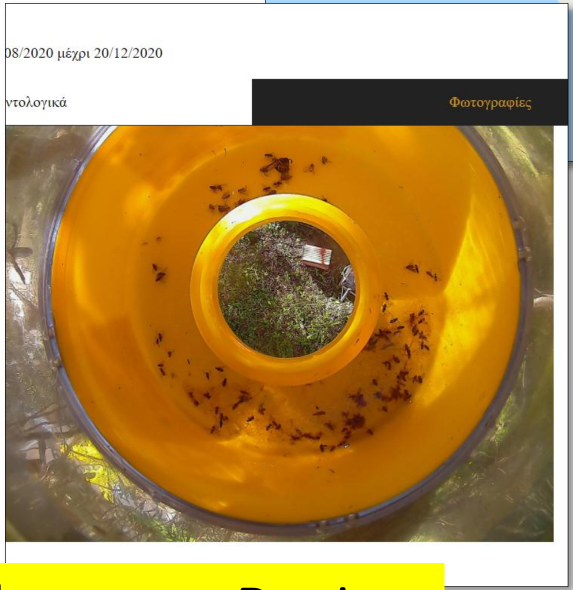
https://enceladus.di.ionio.gr/data/

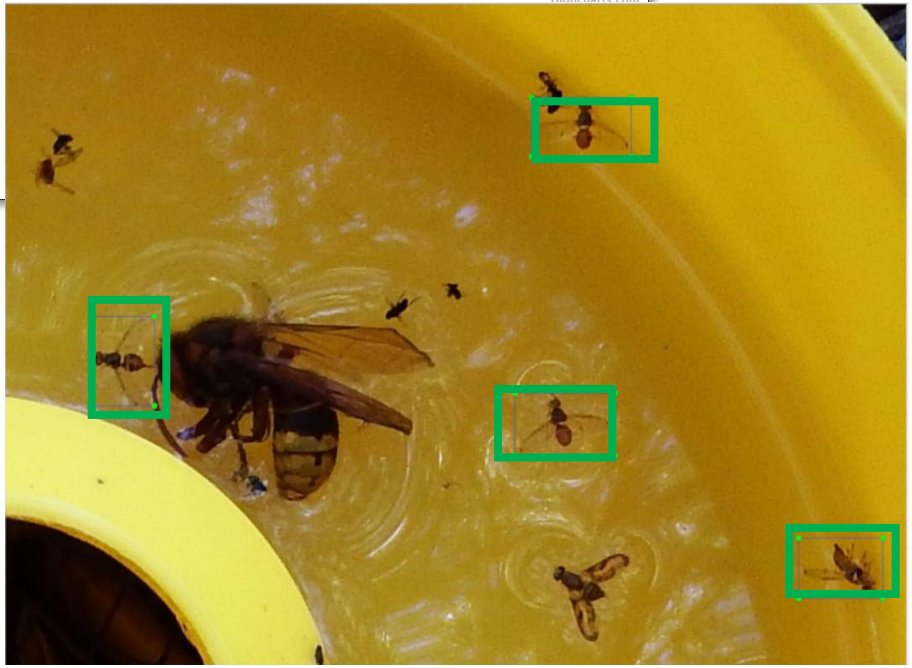


https://enceladus.di.ionio.gr/data/grafimata.php

Computational Modelling Laboratory



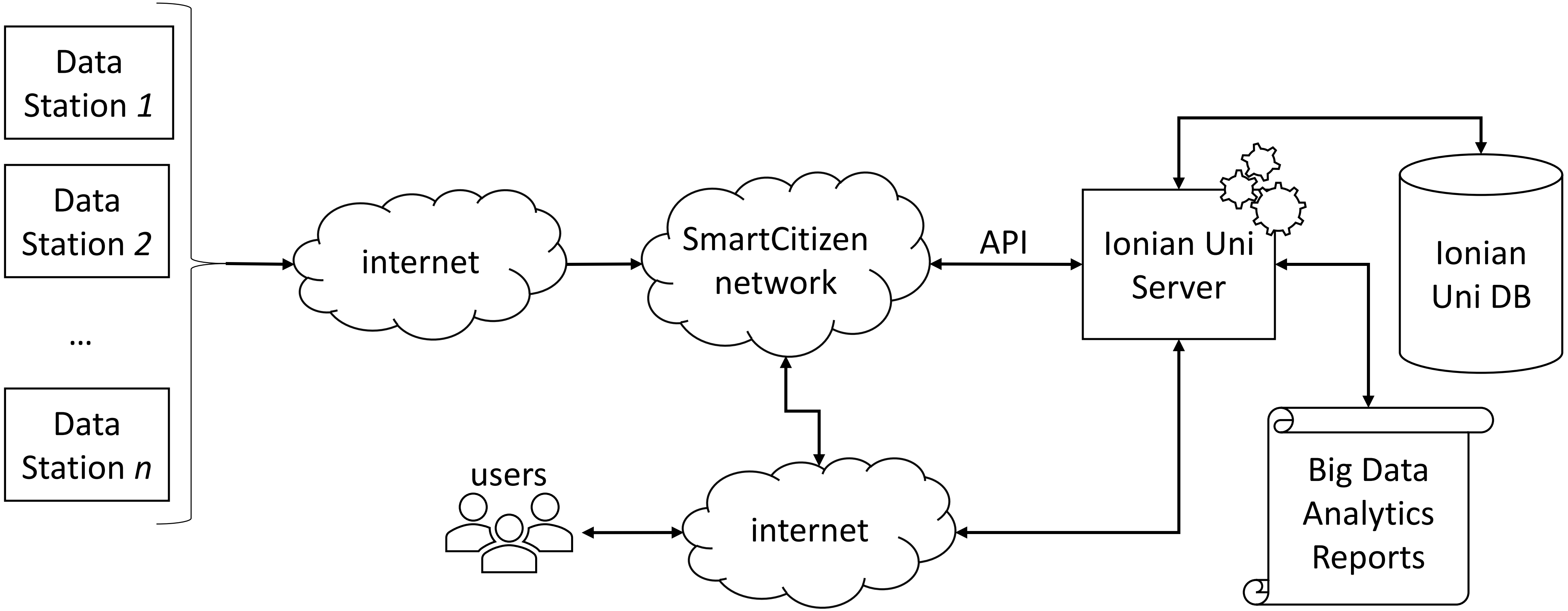




OliveObserver Project

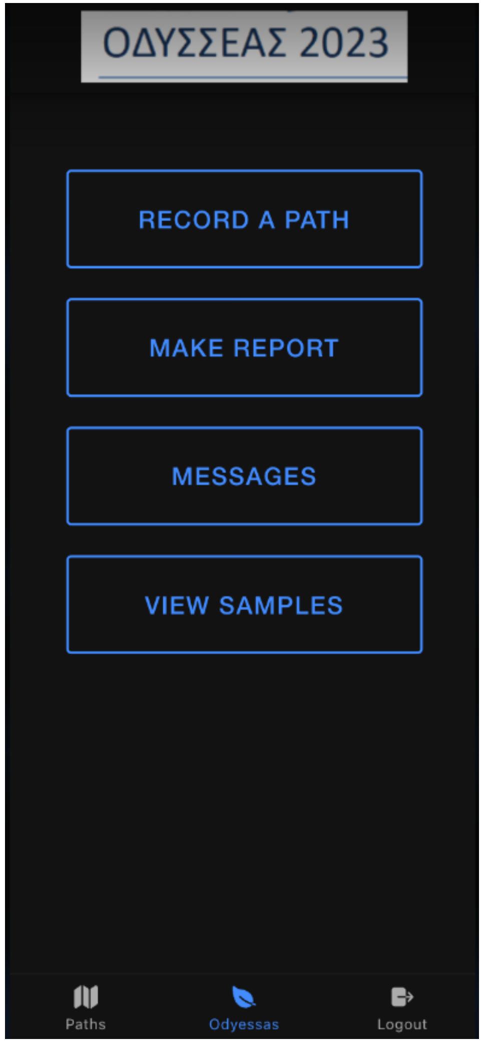
Olive Culture project

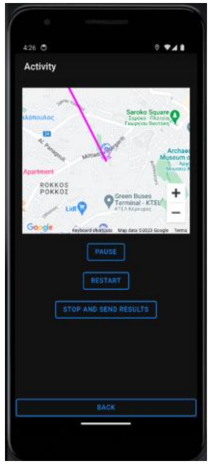
Architecture

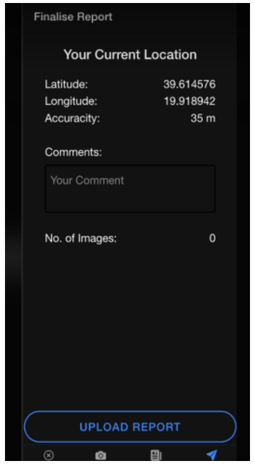


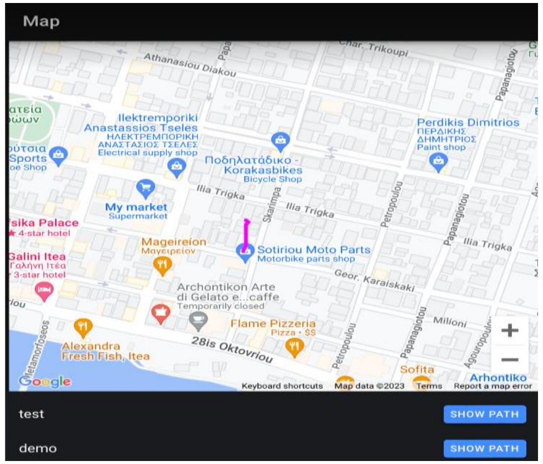
Project Odisseas

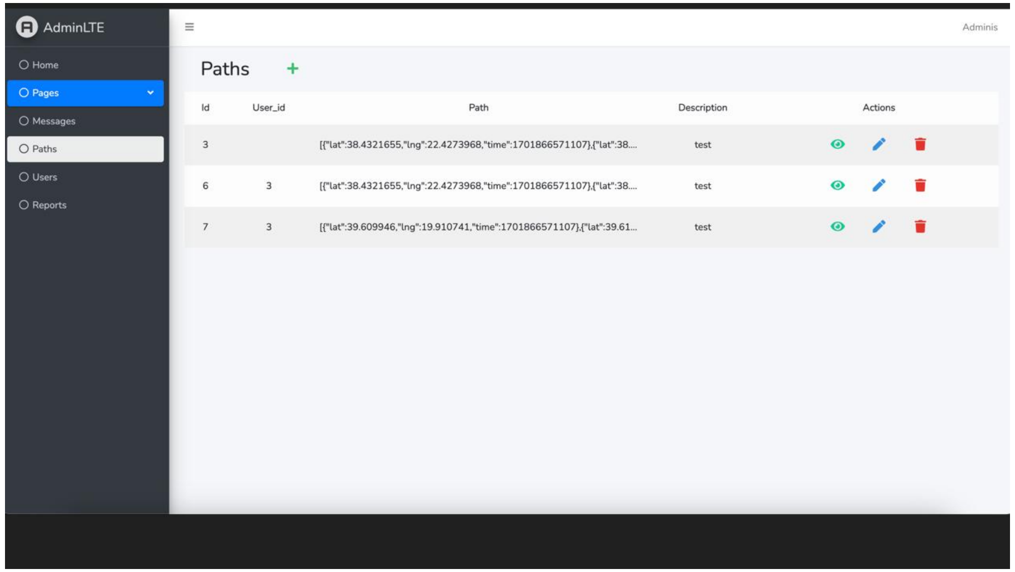
Computational Modelling Laboratory





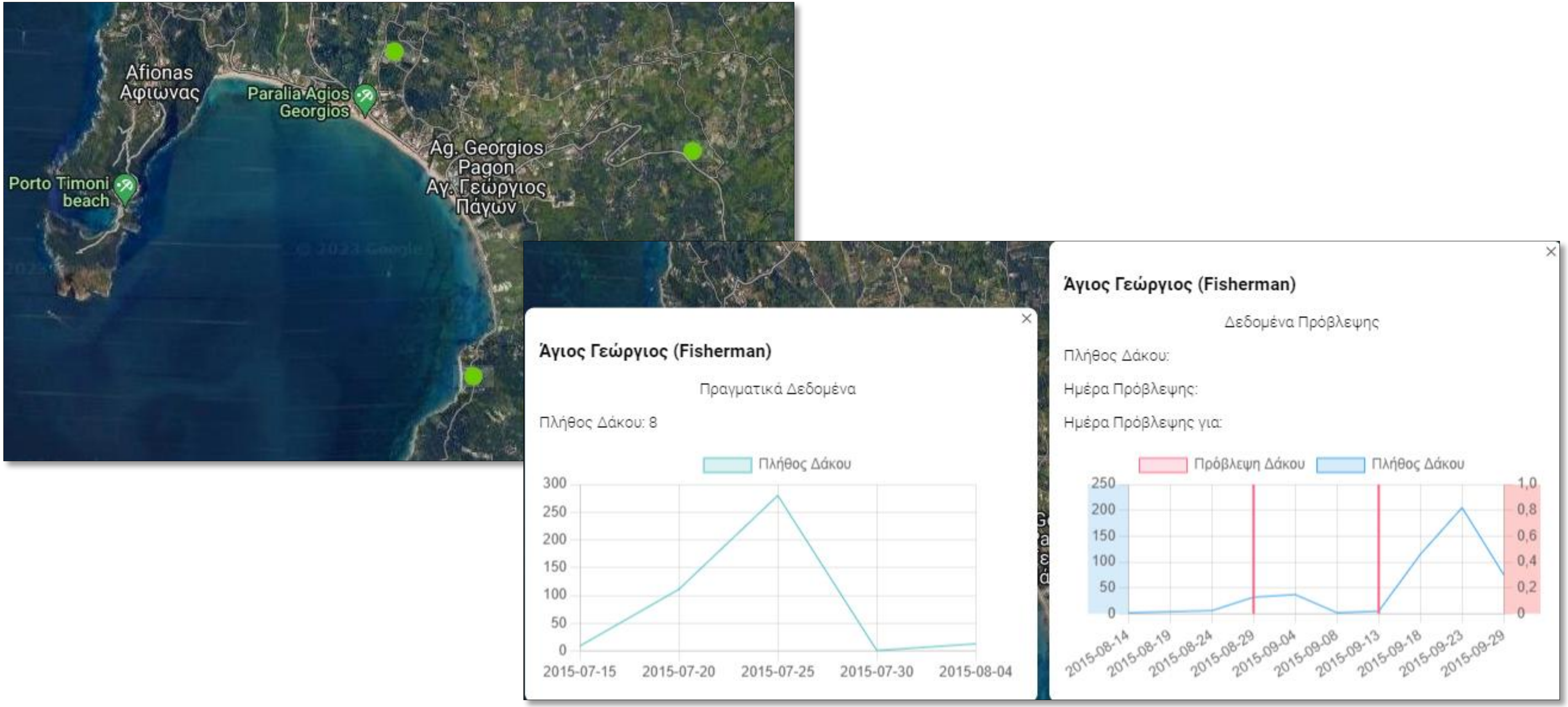






Computational Modelling Laboratory

Project Dakos



Thank you for your attention

Questions?

karydis@ionio.gr

Computational Modelling Laboratory

IMNH
INSTITUTE OF MECHANICS OF NATURAL HAZARDS
IONIAN UNIVERSITY · GREECE