FaST Year 1
Closing meeting

March 30th, 2021
# Agenda - Review of deliverables

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Introduction</td>
<td>5’</td>
</tr>
<tr>
<td>1</td>
<td>Mobile application for farmers</td>
<td>30’</td>
</tr>
<tr>
<td>2</td>
<td>Administration Portal for Paying Agency staff</td>
<td>20’</td>
</tr>
<tr>
<td>3</td>
<td>API for researchers and third-party services</td>
<td>10’</td>
</tr>
<tr>
<td>4</td>
<td>Platform architecture</td>
<td>15’</td>
</tr>
<tr>
<td>5</td>
<td>Documentation (admin portal and technical)</td>
<td>10’</td>
</tr>
<tr>
<td>6</td>
<td>Location of assets</td>
<td>5’</td>
</tr>
<tr>
<td>7</td>
<td>Questions &amp; answers</td>
<td>20’</td>
</tr>
</tbody>
</table>
Context

- The FaST vision is a core platform for the generation & re-use of solutions for sustainable and competitive agriculture
- FaST incorporates space data (Copernicus and Galileo) and other public and private datasets
- With a modular design, FaST supports additional use cases by actors to extend and enhance the platform and the creation and deployment of add on solutions & access of institutional / commercial solutions wanting to connect to FaST

3 DGs

- DG DEFIS
- DG AGRI
- DG DIGIT

4 regions / 3 countries

- Estonia
- Piemonte
- Castilla y León
- Andalucía
Farmers
Improve agronomic performance while reducing fertilizer cost and environmental impact
- mobile app & web app

Advisors
Assist farmers in developing an efficient and compliant NMP
- mobile app & web app

Managing authorities and paying agencies
- User administration
- Gather consolidated data
- 2-way communication
- Geo-tagged photos
- web portal

Policy-makers & researchers
- Develop innovative solutions to optimize the use of inputs in agriculture
- Ensure data security, privacy and anonymization
- web portal, APIs

Institutional & private partners
- Propose innovative and relevant services to the extensive farming user community
- integrate through APIs and opt-in service marketplace

User base of small farmers to propose advanced services

Common platform

Knowledge and trends from consolidated data, large datasets for statistics and CAP strategies

Earth Observation
Sentinel, Copernicus, soil data

Nutrient management
What & when, nutrient budget, best practices

Administrative data
IACS, LPIS, NVZ, Natura 2000

Reduced costs, environmental compliance, time saving, opt-in services

Easy data sharing between farmers and advisors
Benchmarks

Environmental monitoring, increased 2-way comms, economies of scale & CAP digitization

2-way messaging between MA/PA and farmer

User data
Pictures, soil samples

Additional services
Compliance, basic payments, environmental impact, etc.

User administration
Gather consolidated data
2-way communication
Geo-tagged photos

web portal
Timeline

- Onboarding
- On-site workshops
- DIAS selection & contractualisation
- Paying Agencies: development of IACS/authentication APIs
- Core developments: cloud infra, mobile app, admin portal
- Custom developments: data sources, authentication connectors, IACS connectors, fertilization algorithms
- User tests
- Feedback & iterative improvements
- Trainings & documentation

Jan 7th Kick-off (Brussels)

March Covid-19

July 16-17th Half-way webinar

September 18th Candidate MS webinar

October v0.2.3 … Iterative delivery …

Mar 4th v1.0

Mar 30th Year 1 Closing Meeting

FaST Year 1
Covid-19 impact
The pandemic prevented all travel and had a significant impact on a project which is, at its core, a pan-European collaboration

- Travel restrictions started mid-March 2020
- Only the first round of user consultations was performed on site in each region in February 2020
- All following interactions and collaboration were through virtual meetings
  - Approx 150 virtual meetings were held (across all regions)
- All trainings, Q&A sessions and technical workshops were conducted remotely
- Feedback from beta users was collected by the Paying Agencies and reported to the delivery team through the ticketing system
Mobile application
For farmers
Mobile application
A single mobile mobile application for all the regions, configured at runtime

- Compatibility
  - iOS mobile phones
  - Android mobile phones
  - Web browsers (mobile and desktop)

- iOS and Android apps in public beta:
  - https://testflight.apple.com/join/7d0N19Oa

- Web application publicly available:
  - https://app.beta.fastplatform.eu
Authentication

Default login + password + delegation to PA identity providers

- Default FaST identity provider
  - Based on login + password
  - Password recovery workflow
  - Brute force protection (auto-banning)

- Delegation to PA or national identity providers

- Same authentication mechanism(s) on the Administration Portal
Management of campaigns
Campaign-based work environment

- Creation of a new campaign based on:
  - IACS data import
  - Duplication of an existing campaign
  - Demo data (for test users)
- All campaigns are stored and the user can switch between campaigns at any time
- The complete work environment of each campaign is stored and accessible for visualization (parcels, crops, fertilization plans, etc)
- FaST also allows multiple holdings per user (for farm advisors)
Management of campaigns
IACS data import

- Farmers can create a campaign based on IACS data (from the previous season):
  - Get the list of farms that the user is allowed to “see”
  - Get the latest parcels submitted for a farm (ie the latest GSAA): polygons, crops

- Once the data is in FaST, the user can edit it
  - add/remove parcels (parcels can be added from the LPIS/GSAA layer)
  - edit crop and yield
  - etc

- One-way process, the farmer edits are not pushed back to the IACS

- Parcel attributes (soil estimate, constraints,...) are computed dynamically when a parcel is added
Fertilization recommendation

Based on operational algorithms provided by the Member States and ported to FaST

- 4 operational algorithms for fertilization (provided by MS) have been implemented:
  - **Fertilicalc**: NPK & fertilizer product recommendation
  - **Visione**: NPK recommendation including tree crops
  - **VegSyst**: N + irrigation reco for greenhouse crops
  - **ARC**: PK recommendations

- Form-based interfaces with default values when available (soil, yields, etc). All parameters are overridable by the user.
- Administration interface for algorithms
Fertilization recommendation

Export of fertilization plan to pdf and nitrogen legal limits

- Fertilization plan can be downloaded by the user as a pdf (custom format for each algorithm) either at the parcel level or at the campaign level.
- Nitrogen legal limits are displayed at parcel-level depending on plot constraints (NVZ, water body proximity).
- Legal limits are configurable in the Administration Portal.
GIS data sources and maps

- **Integrated:**
  - Nitrate Vulnerable Zones
  - Natura2000 areas
  - Water bodies
  - Soil chemical properties
  - Agricultural parcels (all)

- **Data used for:**
  - Map overlays
  - Estimating soil properties of farmer plots
  - Computing parcel constraints (NVZ, proximity to water body)

- **Custom WMS/TMS data sources with reprojection and server-side cache**
Two-way messaging + broadcasting

- Two-way ticketing system
  - Queries can be initiated by the farmer or by the Paying Agency
  - Queries status (open, resolved, closed, etc)
  - Conversation on each ticket (text + photos)
  - Push notifications on ticket changes
  - All farm members can reply on a ticket

- Broadcasting:
  - One-way communication from PA to all farmers
  - Generates push notifications
  - To be used for general announcements
  - Farmers cannot reply to a broadcast
Temporal series of Sentinel images

- Based on **Sentinel-2 imagery (level 2A)**, from sobloo DIAS catalogue
- **RGB + NDVI**, with navigation between dates
- **Cloud mask** cropped from NDVI + cloud cover indicator
- Configurable historization (currently 12 months) and AOI
- Automatic processing + tiling for faster access from the maps
- Available in the mobile app and in the Administration Portal, in **all the maps**, with farm parcels overlaid
Geolocation + EGNSS4CAP stabilization

- Geolocation provided by:
  - the underlying operating system (iOS and Android app)
  - the standard HTML5 APIs (web app)

- On mobile phones, the OS geolocation is based on the fusion of multiple sensors (including Galileo, GPS, wifi, cell, etc)

- For the geo-tagged photos, on Android only, the geolocation is stabilized using an algorithm provided by the GSA, developed as part of the EGNSS4CAP project
Geotagged photos

- Users can take pictures from within the iOS and Android apps, with their current location displayed on a map.
- Geolocation is stored as well as date/time, heading and elevation.
- Users can add a description and link the photo to a specific parcel.
- Photos are automatically uploaded and retried in case of flaky network.
- Users can view all their photos on a map.
Administration Portal
For Paying Agency staff
Administration Portal
A web-based environment to manage user accesses and data, and configuration parameters

- Exposes the regional FaST data for visualization, edit, add and delete
- Exposes configuration parameters (maps, etc)
- 4 live portals:
  - [https://portal.ee.beta.fastplatform.eu](https://portal.ee.beta.fastplatform.eu)
  - [https://portal.es-cl.beta.fastplatform.eu](https://portal.es-cl.beta.fastplatform.eu)
- Object-based permissions allow to tailor the accessible areas to different PA user types
API
For researchers and third-party services
API access
Remote access to the FaST data store

1. For Paying Agency staff and researchers
   - Access to all the regional FaST data via a secure API
   - Object-types permissions e.g. allow access to farm descriptions but not to user details

2. For third-party services
   - Access to the data of farms that have subscribed to the service
   - Object-types permissions e.g. allow access to farm description but not to user details
API access
A GraphQL endpoint to access the regional FaST data

- GraphQL-based (HTTP requests)
- Exposes the full object graph
- **Read-only** w/ aggregation capabilities
- Authentication through API keys
- **Object-level permissions**
- GraphQL API is auto-discoverable: API schema is exposed by the API itself
- Other types of APIs can be added in the future (modular architecture)

```graphql
fetch data from the table: "fertilization_plan"

**Type**
[fertilization_plan!]

**Arguments**
distinct_on: [fertilization_plan_select_column!]
distinct select on columns
limit: Int
limit the number of rows returned
offset: Int
skip the first n rows. Use only with order_by
order_by: [fertilization_plan_order_by!]
sort the rows by one or more columns
where: fertilization_plan_bool_exp
filter the rows returned
```
All API calls are authenticated through API keys:
  ○ Authorization: ApiKey JxbuioROBqtLkni

API keys are created (and revoked) in the Administration Portal

Object-level permissions are set in the Administration Portal (same as staff users)

Farm-level permissions based on opt-in subscription from the user or through an auto_subscribe parameter
Example of add-on module currently running on FaST

Satellite image tiling module

- Watches the Sentinel L2A catalogue exposed by the DIAS (Sobloo) on region’s AOI
- Loads images, computes NDVI, builds the tiles pyramids
- Serves catalogue entries from its API gateway and tiles from object storage

Authentication is provided by the FaST API gateway, which delegates user requests to the module.

Scalability and high availability are provided by the FaST orchestration layer.
Example of external service connecting to FaST

Example Smart Farming service

- The Smart Farming service gets registered on the regional FaST
- User voluntarily subscribes to the Smart Agriculture service in the FaST app
- User can access the service from the FaST app: he/she is redirected to the designated Smart Farming web page or universal link served by the service
- The service can then query the FaST API on behalf of the user to get his farm data and propose services
- Note: FaST uses a single data model (ontology) across regions, hence the connection/business logic for a specific region is portable to other regions.

Smart Farming service requests an API Key from the regional admin

FaST user subscribes to the Smart Farming service

User clicks on “access Smart Farming service”

User is redirected to the Smart Farming service’s homepage

The Smart Farming service can now query FaST data of the user
Extending and building upon FaST

**FaST add-ons**
- Deployed and running on the FaST platform
- Consumed by the user through the FaST user interfaces
- Example:
  - Fertilization service
  - Other CAP use cases
  - ...

**External add-ons**
- Deployed and running on an external platform / provider
- Consumed by the user through the FaST user interfaces
- Examples:
  - Existing algorithms from other projects (eg H2020)
  - ...

**Third-party services**
- Running outside the FaST platform
- Proposing their own user interfaces
- Users voluntarily subscribe to the service
- The service can access the farmer’s data on the FaST platform (authentication, permissions and GDPR/privacy requirements apply)
Platform architecture
Modularity, scalability, extensibility
FaST High Level Design

**Access layer**
- Farmer client app
- MA/PA portal
- API / export data

**Services:**
- Vegetation status service
- Fertilization service
- Weather service
- Messaging and notification service
- Soil info storage service
- Additional services

**Data**
- Earth observation data
  - Satellite, weather
- Administrative/open data
  - LPIS, NVZ, regulations, etc
- User-generated data
  - Soil sampling, crop rotation, nutrient plans...
- Remote APIs
  - GSAA, weather, etc

**Container orchestration**
- Scalability
- Network
- Distributed storage

**Infrastructure as a Service**
- CPU
- RAM
- HDD

**GitLab**
- CI/CD
- Bazel
- GitOps

**Infrastructure as code**

**Authentication**

**Serving and eventing system**

**Remote APIs**
- Open data
- Administrative/open data
- User-generated data
- Earth observation data
- Earth observation data

**Databases / caches**
- PostgreSQL
- Redis
- HASURA

**Additional services**
- Earth observation data
- Administrative/open data
- User-generated data
- Earth observation data

**Other tools**
- open data
- Administrative/open data
- User-generated data
- Earth observation data

**FaST v1 - Year 1 Closing Meeting**

March 2021
High level architecture: core / custom / add-on blocks

Common/core codebase
(∼80%+ of the codebase is common to all regions)

Custom codebase and data

Add-ons

Container orchestration

Infrastructure as a Service
Modularity, scalability, observability

**Modularity**
- Docker-based deployments
- Each regional deployment is a combination of:
  - Core modules
  - Custom modules
  - Add-ons
- The modules are self-contained and re-usable bricks that can also be ran on other infrastructure with minimal changes

**Scalability**
- The services scale independently and automatically following their current load (up or down)
- Multiple read replicas for databases
- 2 automatic scaling levels:
  - Start/stop new containers (services)
  - Start/stop new nodes (machines)

**Observability**
- The mesh of micro-services is fully observed in real-time to ensure a complete vision of the system in operation
- Security policies are enforced from a central control plane, isolating services and data as necessary
- Constant monitoring and automatic failover / retry
Load increases
Number of services increases

Number of services decreases
Load decreases
Security & Data Privacy
With PwC Cyber Security division

Isolation
- Even though FaST is a single platform, it is split into isolated regional environments
- Isolation between environments is at the network level (control plane), and at the logical level
- Regional data and services are isolated from other regions (i.e. there is no Estonian service that processes Spanish data)

Protection
- Encryption & Authorization:
  - Data encrypted in flight and at rest.
  - Encryption keys managed and rotated by KMS.
- WAF, DDoS protection
- Industry-standard frameworks (Django, Hasura)
- All services require authentication + principle of least privilege
- Service Mesh: centrally controlled network policies

Operations
- CI/CD workflow:
  - Secrets scanning
  - Distroless containers
- MFA for accessing the cloud console

Backups & redundancy
- Redundant instances of all services
- Automatic backups of databases
- Multiple availability zones
Support & Documentation
Website, service desk, user and technical materials
fastplatform.eu website

- Live [https://fastplatform.eu](https://fastplatform.eu) website, including:
  - Project information (participants, what is FaST, who is it for, features, etc)
  - Events and timeline
  - Webinar materials (slide deck and video)
  - Application and Administration Portal screenshots

- Contact form connected to the project service desk
  - 50+ enquiries received over the course of the project
Ticket-based service desk

- Connected to the support@fastplatform.eu email address
- Tracking of incoming requests and resolution
- Linked to the overall project issue tracking system and to the code versioning
- Approx **170 tickets** opened based on regional feedback (deduplicated)
User documentation

- **Documentation for Paying Agency** staff available directly within the Administration Portal
  - In the header bar, on the top-right of the screen
  - Also at [https://gitlab.com/fastplatform/core/-/tree/master/services/web/backend/docs/en](https://gitlab.com/fastplatform/core/-/tree/master/services/web/backend/docs/en)

- **Quickstart guide** developed for mobile application users (one-pager)
  - Available for download at [https://gitlab.com/fastplatform/docs](https://gitlab.com/fastplatform/docs)

- **Testing guide** developed for beta testers

- User documentation is translated in Italian, Estonian, Spanish
User documentation
Administration Portal

- Addresses the context of each feature, as well as “How do I…?” scenarios
- Covers all areas of the Administration Portal:
  - Add-ons & API accesses
  - Imported GIS data
  - Farms / Fertilization plans
  - Maps
  - Messaging
  - Photos
  - Nitrogen legal constraints
  - Managing users (authentication, authorization)
- Search functionality
Technical documentation
Platform architecture, orchestration, security and services

- The technical documentation is located with the source code, as per usual open-source practice
  - **Platform blueprints** (infra, CI/CD, security, etc)
  - **Infrastructure**: cloud resources enrolled in the platform
  - **Orchestration**: Kubernetes orchestration, service mesh
  - **Continuous Integration and Deployment** pipelines
  - **Modules and services**: the main modules and services running on the platform
  - **Authentication and authorization**
  - **Mobile and web applications**: user interfaces
  - **Runbook**: platform monitoring and maintenance guidelines

- Each service also has its own documentation, colocated with the source code of the service
Privacy Policy and Terms & Conditions
Developed for the beta testing phase

- Registered in the Administration Portals of the 4 regions/countries and displayed to every new user since February 2021
- Translated to English, Spanish, Italian and Estonian
- Currently, the Privacy Policy allows data collection and usage only for testing purposes and to improve the FaST service
Journey Documentation Report
Support the potential successive scaling up to other Member State
Journey Documentation Report
Support the potential successive scaling up to other Member States

- **Part 1 - Preparation activities:** workshops, resources, existing project assets
- **Part 2 - Project startup:** onboarding, introduction to the FaST platform, clarifications on MS involvement, what is expected from the MS, and what the MS can expect from the delivery team and the EC, kickstart of all technical streams
- **Part 3 - Iterative development & testing:** main phase of the implementation (customizations, algorithms), PA develop the necessary APIs (if required). Iterative delivery and incremental testing be test users designated by the PA
- **Technical annex**
7 Location of assets
Source code repository
All FaST source code has been open-sourced

- [https://gitlab.com/fastplatform](https://gitlab.com/fastplatform)
- Source code of:
  - All the infrastructure configuration
  - All the orchestration configuration
  - All the CI/CD pipelines
  - All the core and custom services
  - The mobile application
  - The [https://fastplatform.eu](https://fastplatform.eu) website
- All documentation:
  - User documentation for the Administration Portal and quickstart guide for the mobile app
  - Technical documentation, runbook and blueprints
Open-source
Under the MIT license

- The complete FaST codebase is open-source
- FaST is built using only open-source technologies and bricks: no license costs for running FaST

Some exceptions:
  - The security layer / configuration is not made public but available from the repository on demand
  - The VegSyst and Visione algorithms are not open-sourced (request from authors)
  - The Fertilicalc algorithm is open-sourced under the GPL license (hence the FaST implementation of Fertilicalc as well)
Live platform
Common FaST platform currently running on the sobloo DIAS

- **Administration Portal:**
  - [https://portal.ee.beta.fastplatform.eu](https://portal.ee.beta.fastplatform.eu)
  - [https://portal.es-cl.beta.fastplatform.eu](https://portal.es-cl.beta.fastplatform.eu)

- **Mobile application:**
  - iOS version: [https://testflight.apple.com/join/7d0Nl9Oa](https://testflight.apple.com/join/7d0Nl9Oa)
  - Web version: [https://app.beta.fastplatform.eu](https://app.beta.fastplatform.eu)

- **Website:** [https://fastplatform.eu](https://fastplatform.eu)
Questions & answers
Thank you