INFO4AGRO APPLICATION: COMPREHENSIVE DEVELOPMENT & ARCHITECTURE REPORT

EXECUTIVE SUMMARY

Info4Agro is an innovative web-based Decision Support System (DSS) developed within the ALIANCE project (CZ.02.01.01/0.0/0.0/16_040/0000362) as a cloud-hosted platform for precision agriculture. The application integrates:

- **AI-enhanced localized weather forecasting** (FW10010449-V5)
- **Intelligent cloud removal from satellite imagery** (FW10010449-V6)
- **Comprehensive agricultural decision support** through geospatial analysis

The platform connects real-time IoT sensor data, satellite remote sensing, machine learning models, and agronomic knowledge to provide farmers and advisors with precise, location-specific decision support for field operations.

PART 0: HARDWARE INFRASTRUCTURE AND GPU-ACCELERATED COMPUTING PLATFORM

0. Introduction: The Info4Agro Computing Platform as a Pilot Validation Environment

The Info4Agro platform represents not only a software innovation in precision agriculture decision support systems, but also a significant achievement in the operationalization of computationally demanding artificial intelligence models at the edge of practical agricultural deployment. The underlying hardware infrastructure upon which the Info4Agro system operates has been specifically designed and configured to function as a pilot validation environment (poloprovoz in Czech terminology) for GPU-accelerated computing workloads related to weather forecasting enhancement and cloud removal from satellite imagery. This computational infrastructure serves the dual purpose of enabling the intensive model training activities required to develop increasingly accurate prediction and image processing models, while simultaneously supporting the production inference workloads that deliver decision support information to farmers and agricultural advisors in near real-time.

The distinction between the software algorithms and the underlying hardware platform is critical to understanding the true technological achievement of the Info4Agro system. The machine learning models that enable the system's core capabilities—whether the U-Net super-resolution networks for weather downscaling or the UnCRtainTS neural networks for cloud removal—are computationally intensive operations that would be prohibitively slow or entirely impractical to execute on conventional CPU-based computing hardware. The specialized graphics processing units (GPUs) that form the core of the Info4Agro computational infrastructure are fundamentally different from traditional central processing units in their architectural design and are optimized for the highly parallel matrix operations that constitute the mathematical foundation of modern deep learning. This section describes how the Info4Agro hardware platform has been architected, configured, and operationalized to support both the research and development activities that drive continuous improvement of the system's AI models and the production operations that deliver value to end users.

0.1 Cloud Infrastructure Foundation: Lesprojekt Cloud on OpenStack

The Info4Agro computational infrastructure is hosted on the Lesprojekt Cloud, a private cloud computing platform built on the OpenStack open-source cloud infrastructure software. OpenStack was deliberately selected for this application because of its combination of sophisticated resource management capabilities, support for heterogeneous computing resources (including GPU acceleration), proven scalability at enterprise scale, and the flexibility it provides for customization and integration with specialized workloads. The decision to utilize a private cloud infrastructure rather than relying on public cloud providers reflects both the security and data sovereignty requirements inherent in agricultural systems that may process sensitive farm operational data, and the desire to maintain direct control over the computational resource allocation and optimization strategies necessary for efficient operation of the compute-intensive machine learning workloads.

[TODO: Provide specific details about the Lesprojekt Cloud infrastructure including: data center location(s), total compute capacity available, network architecture and bandwidth specifications, redundancy and failover capabilities, current operational status and uptime metrics.]

The OpenStack architecture underlying the Lesprojekt Cloud provides a management layer that abstracts the underlying physical hardware resources and presents them to applications through standardized cloud computing interfaces. This abstraction is particularly valuable for the Info4Agro system because it allows the platform to be agnostic to the specific underlying hardware while still making efficient use of available resources. The management layer provides sophisticated scheduling algorithms that can distribute computational work across available resources, monitor resource utilization in real-time, enforce resource quotas and fair sharing policies, and dynamically provision additional resources when demand increases or deallocate resources when they are no longer needed.

0.2 GPU-Accelerated Computing Architecture

The core of the Info4Agro computational infrastructure consists of specialized computing nodes equipped with graphics processing units optimized for high-performance numerical computing. GPU-accelerated computing has become the de facto standard for training and executing deep neural networks at scale because GPUs provide orders of magnitude improvement in computational throughput compared to conventional CPUs for the specific workloads inherent in machine learning—particularly the dense matrix multiplication and convolution operations that are fundamental to convolutional neural networks (CNNs) and other deep learning architectures used in Info4Agro.

The architectural separation between CPU-based and GPU-based computing workloads is a key design principle of the Info4Agro platform. CPU-based computing resources are provisioned for the operational overhead activities that are not computationally intensive but are necessary for system operation: these include the Flask REST API servers that receive and respond to user requests, the database query engines that retrieve and store system metadata, the data ingestion pipelines that download and process input data from external sources, and the orchestration frameworks that coordinate work across the system. GPU-based resources are reserved for the computationally intensive machine learning workloads that form the core value proposition of the system: specifically, the training of new machine learning models and the execution of inference on these models to generate predictions or processed imagery products.

[TODO: Specify the following GPU infrastructure details:

- GPU model/architecture (e.g., NVIDIA A100, RTX 6000, etc.)
- Number of GPU units available per compute node
- Total GPU memory (VRAM) available per unit
- GPU interconnect technology (NVLink, PCIe, etc.)
- Theoretical peak performance (TFLOPS) of available GPU resources
- Current GPU utilization metrics and typical allocation patterns]

The GPU computing environment is configured to support the primary deep learning frameworks and libraries used in the Info4Agro AI components. The weather forecasting model training utilizes frameworks such as TensorFlow and PyTorch, which are the predominant open-source libraries for deep learning research and production systems. These frameworks have been specifically optimized by their developer communities to extract maximum performance from available GPU hardware, implementing sophisticated kernel fusion strategies, mixed-precision arithmetic to reduce memory consumption and communication overhead, and distributed training capabilities that allow individual training jobs to utilize multiple GPUs in parallel for accelerated training of large models.

[TODO: Provide specific information about:

- CUDA version and compatibility
- Deep learning framework versions deployed

- Data loading and preprocessing pipeline optimization strategies
- GPU memory management and optimization techniques in use
- Monitoring and profiling tools used to identify performance bottlenecks]

0.3 Computational Workload Categories and GPU Utilization Patterns

The GPU infrastructure supports two fundamentally different computational workload patterns that have distinct performance characteristics, resource requirements, and operational considerations: training workloads and inference workloads.

Training workloads involve the iterative process of exposing a neural network model to labeled training examples, computing the difference between the model's predictions and the ground truth labels, and using this difference signal to adjust the model's internal parameters through backpropagation. These training workloads are computationally intensive and typically run for extended periods (hours to days for the larger models used in Info4Agro), making heavy utilization of available GPU memory and compute capacity. The training of the U-Net super-resolution models used for weather downscaling represents a significant portion of the training workload on the Info4Agro infrastructure. These models are trained on historical pairs of global forecast data (at coarse resolution) and high-resolution reference data (from ERA5-Land reanalysis), allowing the model to learn the relationship between coarse input and high-resolution output. Additionally, the per-station machine learning models (CatBoost, LSTM, MLP variants) used for localized weather prediction require periodic retraining as new observational data accumulates, allowing the models to learn any systematic biases or drifts that may develop over time.

[TODO: Provide specific metrics about training workloads:

- Typical model training times and iteration patterns
- GPU memory requirements for different model architectures
- Batch size and data loading throughput requirements
- Frequency of retraining cycles for production models
- GPU utilization rates during typical training sessions]

The cloud removal and gap-filling workloads represent another major category of training activity on the Info4Agro GPU infrastructure. The UnCRtainTS neural network models that perform cloud removal and vegetation index recovery are trained on historical Sentinel-2 satellite imagery paired with cloud-free reference data, allowing the models to learn to predict what the satellite would observe if clouds were not present. These models are complex convolutional architectures with significant depth and parameter counts, requiring careful management of GPU memory through techniques such as gradient checkpointing and mixed-precision training. The training process for satellite image processing models is particularly data-intensive, requiring efficient loading and preprocessing of large raster imagery files from storage systems.

[TODO: Document:

- Training datasets sizes for satellite image models
- Preprocessing pipelines and data augmentation strategies
- Memory-bandwidth bottlenecks and optimization strategies
- Multi-GPU training orchestration for large models
- Model checkpointing and recovery procedures]

Inference workloads involve applying already-trained models to new input data to generate predictions or process results. Unlike training workloads, inference can often be performed with significantly lower GPU memory requirements (as only forward passes through the network are required, not the backpropagation computations that drive memory consumption during training), but inference workloads must complete within strict latency requirements to provide timely decision support to users. The Info4Agro platform performs inference workloads both synchronously (when a user requests a forecast or satellite product immediately) and asynchronously (when forecasts are pre-computed at regular intervals for efficient caching and rapid delivery to end users).

The weather forecasting inference workload involves executing the trained U-Net super-resolution model on incoming global forecast data, followed by execution of the station-specific prediction models (CatBoost, LSTM, MLP) to generate localized weather predictions. These inference operations must be performed regularly (typically every six hours when new global forecast cycles become available) for hundreds of meteorological stations distributed across the region. The overall inference latency requirement for the complete forecasting pipeline (global model execution plus per-station localization) should not exceed [TODO: specify latency requirement in hours/minutes], to allow timely delivery of updated forecasts to users and downstream applications.

The satellite image processing inference workload involves applying the trained UnCRtainTS models to newly acquired Sentinel-2 satellite scenes to remove or mitigate the effects of cloud cover. These inference operations are triggered when new satellite imagery becomes available (Sentinel-2 typically revisits any given location every 5-10 days depending on latitude), and the processing can require substantial GPU compute resources because satellite scenes are large (covering areas of hundreds of square kilometers at 10-meter resolution) and must often be processed at full spatial resolution to generate final deliverable products.

[TODO: Specify inference performance requirements:

- Latency requirements for weather forecast inference (end-to-end)
- Throughput requirements (number of stations processed per time period)
- Latency requirements for satellite image processing
- Typical scene processing times at full resolution
- GPU utilization targets and efficiency metrics]

0.4 Job Queue and Workload Orchestration

Because the Info4Agro system must handle both time-critical operational inference workloads and longer-running training and batch processing activities, the GPU infrastructure implements a sophisticated job queue and workload orchestration system that manages the scheduling and execution of different workload types. This orchestration layer is critical for ensuring that the limited GPU resources are allocated efficiently across competing demands and that the overall system maintains responsiveness to user requests while also making progress on longer-running computations.

The job queue system implements priority-based scheduling, allowing time-critical inference operations to be prioritized over longer-running batch operations. When a user requests a real-time forecast or satellite product processing result, the system can identify this as a high-priority job and potentially preempt lower-priority batch operations to ensure rapid completion. Background retraining of models and batch processing of accumulated satellite imagery can be scheduled for times when resource demands are lower (typically overnight or other off-peak periods for the agricultural use cases) to avoid interfering with operational inference workloads that directly serve decision-making activities.

[TODO: Document the job queue system including:

- Queue management architecture and implementation
- Priority levels and scheduling policies
- Job state machine and lifecycle management
- Resource allocation and fairness algorithms
- Monitoring and alerting for queue health and performance
- Integration with OpenStack resource management]

The orchestration of distributed training across multiple GPUs represents a specialized capability of the job queue system. Some of the larger models used in Info4Agro (particularly the UnCRtainTS networks for cloud removal) may benefit from training across multiple GPUs in parallel, requiring sophisticated synchronization of gradient updates across devices and careful management of communication bottlenecks between GPUs. The orchestration system must support both single-GPU training (for smaller per-station models) and multi-GPU distributed training (for larger global models) transparently, managing the complexity of distributed training abstractions.

0.5 Storage and Data Movement

The GPU-accelerated computing infrastructure must be integrated with appropriate storage systems to support efficient data movement between storage and GPU memory. The bottleneck in many deep learning applications is not the raw compute capacity of GPUs but rather the rate at which training data or inference inputs can be loaded from storage into GPU memory. If the data loading pipeline cannot keep the GPUs supplied with data at the rate they can consume it, then the GPUs will sit idle waiting for data, resulting in poor hardware utilization and increased time to solution.

The Info4Agro platform addresses this challenge through several complementary strategies. First, the platform maintains hot data caches in high-speed storage (such as local NVMe solid-state drives on GPU compute nodes) for the most frequently accessed datasets, reducing the need to load data across network connections from centralized storage. Second, the data loading and preprocessing pipeline has been optimized to overlap I/O operations with GPU computation, allowing data for the next training iteration to be loaded while the current iteration is executing on the GPU. Third, the platform implements intelligent data prefetching strategies that anticipate which data will be needed in the future and proactively load it before it is requested.

[TODO: Specify storage architecture including:

- Storage system types (NFS, S3, local NVMe, etc.)
- Storage capacity and throughput specifications
- Data locality strategies and caching policies
- Network bandwidth between storage and compute nodes
- Data staging and migration workflows
- Storage backup and disaster recovery procedures]

For the satellite imagery processing workloads, the data movement challenge is particularly acute because Sentinel-2 data comprises large multi-band raster files (individual scenes can be gigabytes in size), and processing may require loading multiple scenes worth of data into GPU memory for temporal fusion operations. The platform implements specialized data loading strategies for raster imagery, including on-the-fly decompression of compressed satellite data, tiling strategies to break large scenes into GPU-memory-sized chunks, and intelligent resampling to avoid unnecessary data movement when output resolution is lower than input resolution.

0.6 Info4Agro as a Pilot Validation (Poloprovoz) Environment

The Info4Agro hardware infrastructure should be understood in the context of the Czech funding agency's definition of "poloprovoz" (pilot validation environment) as articulated in the research outcome definitions document. According to this definition, a poloprovoz is a validation of the functional capabilities of laboratory procedures at scales larger than pure laboratory settings but still in experimental and validation contexts, specifically before introduction into full production operation at manufacturing or service scales.

The Info4Agro GPU computing platform fits this definition precisely. The underlying hardware configuration represents an experimental and validation environment where AI models for weather forecasting and cloud removal have been developed, trained, and extensively validated in simulated operational conditions before being deployed into genuine production scenarios at agricultural scale. The system is not yet operating at its maximum potential scale (there is room to add additional GPU compute nodes, expand storage capacity, and increase the number of stations or geographical regions served), but rather represents the current production-capable implementation of the research and development activities that have created the underlying AI methodologies.

This characterization as a poloprovoz environment has several important implications for the system's design and operation. First, it acknowledges that the system is operating in a controlled, experimental context where careful monitoring, validation, and measurement of system performance metrics is ongoing. The Info4Agro development and operational teams maintain detailed logs of all model training activities, inference operations, and system performance metrics, enabling continuous evaluation of whether the system is achieving its intended objectives and identification of areas for improvement. Second, it indicates that the current configuration is not necessarily the final optimized configuration—the hardware infrastructure can evolve and be scaled based on lessons learned during this validation phase. Third, it establishes a clear delineation between the current pilot operations and potential future introduction of similar systems into full production agricultural operations at much larger scales.

The validation activities undertaken on the Info4Agro hardware platform serve multiple purposes. From a technical standpoint, the system is validating whether the AI methodologies developed in the research phase can indeed function reliably and accurately in operational conditions with real data, real time constraints, and real system failures and irregularities. From a user perspective, the system is validating whether the decision support products generated by the AI models are actually useful to farmers and advisors in practice, leading to better agricultural decisions and outcomes. From a technological standpoint, the system is providing a testbed for exploring optimizations and refinements to the underlying AI algorithms and system architectures.

1. DEVELOPMENT TIMELINE & PROJECT EVOLUTION

Project Phase Structure (2021-2025)

Based on the document versioning and milestones, the Info4Agro development follows this timeline:

Phase 1: Concept Development (2021-2022) - V1 Final concept

Key Milestones:

- **Q1 2022**: Architectural framework established
  - Definition of overall system concept integrating 5 core modules
  - Identification of data sources and integration requirements
  - UX/GUI strategy development
- **Key Deliverables**:
  - Complete system architecture (SensLog, LocalizedForecastAI, AI EO, Repository, FrontEnd)
  - Data flow specifications between modules
  - Cloud infrastructure design (Lesprojekt Cloud on OpenStack)
  - Security and data protection framework

Target Technologies Identified:

- OpenStack cloud infrastructure
- Machine learning frameworks (TensorFlow, PyTorch)
- PostgreSQL + PostGIS spatial database
- OGC standards compliance (WMS, STAC)

Phase 2: Core AI Component Development (2022-2023)

Parallel Tracks:

Track A: Weather Forecast AI (V5)

- **Q2-Q3 2022**: Methodology development
  - GFS, ERA5-Land, and local station data integration
  - U-Net super-resolution model for ERA5-Land approximation
  - Per-station CatBoost, LSTM, MLP model families
- **Q4 2022**: Initial validation
  - Training on 2020-2022 data
  - Testing on 2023 data (time-consistent split)
  - MAE/MSE metrics baseline establishment
  - Testing on 219 stations in cooperation with Vitispector/EKOVN

- **Key Achievement**: Achieved 24-30% error reduction vs. baseline GFS predictions

Track B: Cloud Removal AI (V6)

- **Q3-Q4 2022**: Methodology validation
    - UnCRtainTS neural network architecture evaluation
    - Sentinel-1 + Sentinel-2 data fusion testing
    - Cloud masking and SCL (Scene Classification Layer) optimization
    - Testing in Czech Republic and Colombia (penositelnost verification)
- **Key Findings**:
    - Mono-temporal variant sufficient for operational deployment
    - MAE for NDVI < 0.05 achieved in clear weather
    - Hybrid approaches (TabPFN-KNN) showed 24% error improvement

Phase 3: Software Implementation & Integration (2023-2024) - V5 & V6

Software Release Strategy:

FW10010449-V5: Localized Weather Forecast Software

- **Deliverable Type**: R - Software
- **Release Timeline**: Q2-Q3 2024
- **Key Components**:
    - SensLog connector for GFS/Open-Meteo ingestion
    - LocalizedForecastAI inference engine
    - Forecast REST API (OpenAPI 3.0.3)
    - Per-location model management in AI PreTrained Model HUB

Critical Operational Milestones:

- Q2 2024: Open-Meteo integration pilot (single station testing)
- Q3 2024: API contract stabilization (breaking changes management)
- Q4 2024: Multi-station scaling decision

FW10010449-V6: Cloud Removal & Sentinel-2 Gap-Filling

- **Deliverable Type**: R - Software
- **Release Timeline**: Q3-Q4 2024
- **Key Components**:
    - EO Data API (OpenAPI/Swagger)
    - Job-based orchestration for long-running processes
    - Vegetation index computation (NDVI, EVI, SAVI, VCI, TCI, NDSI, ALBEDO)

    - STAC catalog integration

Operational Milestones:

- Q3 2024: Core job management framework
- Q4 2024: Index computation pipeline completion
- Q1 2025: STAC standardization and cleanup routines

Phase 4: Platform Integration & Frontend (2023-2025)

Frontend Development (FrontEnd component):

- **Strategy Phase**: UX/GUI analysis and information architecture design
- **Prototyping**: Wireframes and high-fidelity prototypes (Crop Interface, Fertilization Interface examples)
- **Implementation**: Component development, user testing, iterative refinement
- **Current Status**: Operational with continuous UX improvements

Key Frontend Milestones:

- Q4 2023: Dashboard and project management interface
- Q1 2024: Sensor data visualization and time-series charting
- Q2 2024: Satellite image management interface
- Q3 2024: Fertilization map generation and export
- Q4 2024: Planning tools (Place, Crop, Ag-co) full implementation

Phase 5: Production Deployment & Scaling (2024-2025)

Current Status: Advanced Implementation

- **Weather Forecast**: Operational on 219+ stations, integration with ICON-EU underway
- **Cloud Removal**: Production-ready, deployed for Czech regions and pilot areas
- **Frontend**: Fully operational with user community feedback integration
- **Cloud Infrastructure**: Lesprojekt OpenStack with GPU support for model training

---

2. HIGH-LEVEL ARCHITECTURE OF INFO4AGRO

2.1 System Architecture Overview

Info4Agro is built on a **modular, cloud-native architecture** with clear separation of concerns:

text

```
┌─────────────────────────────────────────────────────────────────────┐
│                  INFO4AGRO PLATFORM (Web Application)               │
├─────────────────────────────────────────────────────────────────────┤
│                                                                     │
│  ┌──────────────────┐  ┌──────────────────┐  ┌──────────────────┐   │
│  │   Frontend       │  │   REST APIs      │  │   STAC Catalog   │   │
│  │   (Vue.js/       │  │   (OpenAPI)      │  │   (Interop)      │   │
│  │    Angular)      │  │                  │  │                  │   │
│  └──────────────────┘  └──────────────────┘  └──────────────────┘   │
│            │                     │                     │            │
├─────────────────────────────────────────────────────────────────────┤
│            │  INTEGRATION LAYER                        │            │
│            │  (Data Orchestration & Caching)           │            │
├─────────────────────────────────────────────────────────────────────┤
│            ▼                                           │            │
│  ┌───────────────────────────────────────────────┐     │            │
│  │        CORE PROCESSING MODULES                │     │            │
│  │                                               │     │            │
│  │  ┌──────────────────┐  ┌─────────────────┐    │     │            │
│  │  │ SensLog          │  │ LocalizedFore-  │    │     │            │
│  │  │ (Data Hub)       │  │ castAI          │    │     │            │
│  │  │                  │  │ (Weather ML)    │    │     │            │
│  │  │ • Sensor Data    │  │                 │    │     │            │
│  │  │ • Forecast Cache │  │ • U-Net SR      │    │     │            │
│  │  │ • Coordination   │  │ • CatBoost      │    │     │            │
│  │  │                  │  │ • LSTM/MLP      │    │     │            │
│  │  └──────────────────┘  └─────────────────┘    │     │            │
│  │                                               │     │            │
│  │  ┌──────────────────┐  ┌─────────────────┐    │     │            │
│  │  │ AI EO            │  │ Repository      │    │     │            │
│  │  │ (Cloud Removal)  │  │ (Data Storage)  │    │     │            │
│  │  │                  │  │                 │    │     │            │
│  │  │ • UnCRtainTS NN  │  │ • S3/NFS        │    │     │            │
│  │  │ • Gap-filling    │  │ • PostgreSQL    │    │     │            │
│  │  │ • Index Compute  │  │ • PostGIS       │    │     │            │
│  │  └──────────────────┘  └─────────────────┘    │     │            │
│  │                                               │     │            │
│  └───────────────────────────────────────────────┘     │            │
├─────────────────────────────────────────────────────────────────────┤
│             CLOUD INFRASTRUCTURE                       │            │
│             (Lesprojekt Cloud - OpenStack)             │            │
│                                                        │            │
│  ┌────────────────────┐  ┌────────────────────┐        │            │
│  │ CPU Instances      │  │ GPU Instances      │        │            │
│  │ (Flask, APIs)      │  │ (ML Training)      │        │            │
│  └────────────────────┘  └────────────────────┘        │            │
│                                                        │            │
│  ┌───────────────────────────────────────────────┐     │            │
│  │ Storage & Databases                           │     │            │
│  │ • PostgreSQL + PostGIS (Metadata)             │     │            │
│  │ • NFS/S3 (SAFE products, Rasters)             │     │            │
│  │ • Job Queue (Long-running operations)         │     │            │
│  └───────────────────────────────────────────────┘     │            │
│                                                        │            │
└─────────────────────────────────────────────────────────────────────┘
                                                         │
                         ┌───────────────────────────────┘
                         ▼
                ┌─────────────────────┐
                │ External Services   │
                ├─────────────────────┤
                │ • Copernicus CDSE   │
                │ • GFS (NOAA)        │
                │ • Open-Meteo        │
                │ • ERA5-Land         │
                │ • EUMETSAT LSA      │
                └─────────────────────┘
```

2.2 Modular Component Breakdown

Module 1: SensLog (Data Hub)

Purpose: Central data repository and orchestration layer **Responsibilities**:

- IoT sensor data ingestion from local weather stations
- External forecast data (GFS, Open-Meteo, ICON-EU) management
- Forecast caching and query API
- Data quality control and harmonization
- Time-series data storage with geospatial indexing

Key Subcomponents:

- **SensLog Feeder**: Receives measurements from local stations via SensorObservation API
- **SensLog Connector**: Periodic download of global forecasts with intelligent caching

- **SensLog Orchestrator**: Coordinates data flow to ML modules and receives computed results
- **ISensorObservation**: API for local measurement query
- **IForecastSensor**: API for forecast storage and publication

Module 2: LocalizedForecastAI (Weather Forecast Engine)

Purpose: AI-enhanced hyperlocal weather prediction
Responsibilities:

- Super-resolution of global GFS data to ERA5-Land-like resolution (U-Net)
- Per-station localized prediction (CatBoost, LSTM, MLP)
- Model training, validation, and versioning
- 24-48 hour forecast generation
- Seamless integration into forecast publication pipeline

Key Algorithms:

- **Phase 1**: U-Net encoder-decoder with 4-channel input (temperature, dewpoint, U/V wind)
- **Phase 2**: Station-specific ensemble (CatBoost primary, LSTM/MLP alternatives)
- **Feature Selection**: SHAP-based importance ranking
- **Validation**: Time-consistent train/test splits (2020-2022 training, 2023 validation)

Module 3: AI EO (Cloud Removal & Gap-Filling)

Purpose: Continuous vegetation monitoring despite cloud cover
Responsibilities:

- Cloud detection and pixel-level masking
- Sentinel-2 data reconstruction (gap-filling)
- Vegetation index computation (NDVI, EVI, SAVI, VCI, etc.)
- True Color Image (TCI) generation
- Job-based processing orchestration

Key Algorithms:

- **UnCRtainTS**: Convolutional neural network with attention for cloud removal
- **Mono-temporal variant**: Independent processing of time steps (operational choice)
- **Tiling & Sliding Window**: Mitigation of edge artifacts in large scenes
- **Multi-source fusion**: Sentinel-1 (radar) + Sentinel-2 (optical) + ancillary data

Module 4: Repository (Data Storage & Catalog)

Purpose: Persistent storage and discovery of processed data **Responsibilities**:

- SAFE product archive management
- Processed raster storage (indices, True Color)
- Metadata management via PostgreSQL+PostGIS
- STAC API implementation for interoperability
- Cloud-free scene registry

Storage Hierarchy:

- **Raw Data**: Sentinel-1/-2 SAFE products (NFS/S3)
- **Derived Products**: Vegetation indices (GeoTIFF format)
- **Metadata**: PostgreSQL relational with spatial extension
- **Preview**: PNG thumbnails for quick visualization

Module 5: FrontEnd (User Interface & Decision Support)

Purpose: User-facing application for decision support **Responsibilities**:

- Project and field management
- Interactive map-based visualization
- Sensor data display with time-series charting
- Satellite image management interface
- Fertilization map generation (VRA)
- Climate-based planning tools (Place, Crop, Ag-co)
- Export functionality (GeoJSON, Shapefile, vehicle formats)

Technology Stack:

- **Client**: Vue.js/Angular (modern SPA framework)
- **Visualization**: Leaflet + Mapbox (interactive mapping)
- **Charting**: Chart.js or similar (time-series graphs)
- **Backend**: Flask/Node.js REST API

INFO4AGRO: COMPREHENSIVE TUTORIAL

Step-by-Step Guide with Screenshots

Complete Walkthrough of a Precision Agriculture Platform

TABLE OF CONTENTS

INTRODUCTION & APPLICATION OVERVIEW

Info4Agro is a web-based Decision Support System (DSS) for precision agriculture that integrates:

- Real-time IoT sensor data
- Satellite remote sensing imagery
- Climate and weather modeling
- Crop-specific agronomic data
- Variable Rate Application (VRA) fertilization mapping

The platform helps farmers and agronomists make data-driven decisions for optimal field management, crop selection, and seasonal planning.

STEP 1: ACCESSING YOUR PROJECTS [info4agro]

Upon logging into Info4Agro, you'll see your project dashboard displaying all your farm projects.

What you'll see:

- **Project List Table** with columns:
  - **No.** - Project identifier
  - **Title** - Project name

    ○ **Fields** - Number of individual field parcels

Example Projects:

- VetFarm (30 fields) - Primary farm
- Předenice (58 fields) - Larger operation
- Předenice 2025 (58 fields) - Next season planning

Key UI Elements:

- Left sidebar with navigation menu organized in three sections:
    - **MANAGEMENT** - Project admin
    - **CURRENT PROJECT** - Active project features
    - **PLANNING** - Seasonal planning tools
- Center panel showing interactive map
- Three menu buttons at top (Add project, more options, Exit)

STEP 2: VIEWING PROJECT INFORMATIONb info4agro

Click **"Info"** under the CURRENT PROJECT section to view metadata about the active project.

What you'll see:

text

```
┌─────────────────────────────────────┐
│ CURRENT PROJECT                     │
├─────────────────────────────────────┤
│ ACTIVE PROJECT:    vetfarm          │
│ FROM_DATE:       01. 05. 2024       │
│ TO_DATE:         11. 05. 2024       │
│                                     │
│ [Edit project button]               │
└─────────────────────────────────────┘
```

Important Notes:

- The date range (FROM_DATE to TO_DATE) controls which satellite images are downloaded
- This timeframe affects all vegetation index calculations
- Should match your project's growing season or specific analysis period

Actions:

- Click **"Edit project"** to modify dates, field boundaries, or project metadata

---

STEP 3: EXPLORING SENSOR DATA [screenshots 3, 4, 5, 6]

Click **"Sensors"** to visualize real-time and historical data from meteorological stations.

3.1 Initial Sensor View [youtube]

What you'll see:

- Empty right panel: "No sensor units selected"
- Instructions: "To proceed select at least one sensor unit"
- Map showing all available sensor locations (orange triangles)
- Units dropdown selector
- Comparison Mode checkbox

3.2 Selecting a Sensor Unit [info4agro]

How to select:

- Click on any orange triangle marker on the map, OR
- Use the "Units" dropdown to search for a specific sensor

After Selection:

- Sensor name appears at top (e.g., "Meteo Osek 2 (0.8_L_0044619)")
- Blue highlight on selected sensor
- Right panel shows all available measurements:
    - Air thermometer
    - Anemometer (wind)
    - Barometer (pressure)
    - Hygrometer (humidity)
    - Precipitation
    - Solar radiation
    - Strike distance (lightning)
    - Voltmeter (battery)

3.3 Selecting a Measurement Parameter [info4agro]

How to:

1. Expand a measurement category (e.g., "Air thermometer")
2. Check the checkbox next to the parameter you want to visualize
3. System begins: "Generating charts!"

Available Parameters (Example: Air thermometer):

- Atmos41 air temp
- VP3 air temp 03

3.4 Viewing Sensor Data Chart [info4agro]()

Chart Display Features:

```text
┌─────────────────────────────────────────────────────────┐
│ Air thermometer (1)                                     │
├─────────────────────────────────────────────────────────┤
│                                                         │
│ TIME RANGE CONTROLS:                                    │
│ [1H] [24H*] [1W] [1M] [📅 Calendar]                     │
│                                                         │
│  ┌───────────────────────────────────────────────────┐  │
│  │                                                   │  │
│  │ Temperature Trend (24-hour view)                  │  │
│  │    12:00 → 15:00 → 18:00 → 21:00 → 00:00 → 09:00  │  │
│  │ ╱╲___╱╲___╱╲___╱╲___╱╲___╱╲___╱╲___╱╲___╱╲___     │  │
│  │                                                   │  │
│  └───────────────────────────────────────────────────┘  │
│                                                         │
│ STATISTICS TABLE:                                       │
│ ├─ Min:     1.00 °C                                     │
│ ├─ Max:     12.10 °C                                    │
│ └─ Average: 6.47 °C                                     │
│                                                         │
│ OPTIONS:                                                │
│ [ ] Enable data forecast                                │
│ [ ] Enable global data                                  │
└─────────────────────────────────────────────────────────┘
```

Time Range Options:

- **1H** - Last hour (resolution: minutes)
- **24H** - Last 24 hours (default, resolution: hourly)
- **1W** - Last week (resolution: daily)
- **1M** - Last month (resolution: daily)
- 📅 - Custom date range picker

Advanced Features:

- **Data Forecast:** Shows predicted values for next period
- **Global Data:** Compares with global historical/model data
- **Statistics:** Min, Max, Average calculations

3.5 Comparison Mode [Optional Advanced Feature]

To Compare Multiple Sensors:

1. Check **"Comparison Mode"** checkbox
2. Use SHIFT+click on the map to select multiple sensors
3. The same parameter is displayed from all selected sensors
4. Graphs are synchronized to same time range
5. Hover over lines to identify which sensor each represents

Use Cases:

- Compare temperature at different field locations
- Validate sensor readings between units
- Identify microclimatic variations

STEP 4: MANAGING SATELLITE IMAGERY [screenshots 7, 8]

Click **"Sat. images"** to access satellite image processing tools.

4.1 Accessing Satellite Image Management info4agro

Initial View:

- If no images processed: "Nothing here yet!"
- Message: "This project needs to be processed first"
- Button: "Load location"

After Clicking "Load location":
Two tabs appear:

1. **Manage satellite images** (primary)
2. **Image products** (derived products)

4.2 Managing Satellite Images info4agro

Interface Elements:

text
```
┌─────────────────────────────────────────────────────────────┐
│ MANAGE SATELLITE IMAGES                        │
├─────────────────────────────────────────────────────────────┤
│                                    │
```

```
│ Date range: May 1, 2024 - May 11, 2024  [📅]          │
│ Product: [True color image ▼]                │
│                                   │
│ [👁 Usable] [👁 Hidden]                        │
│ Images that will be used in analysis. Select to hide. │
│                                   │
│ ┌──────────────────────────────────────────────────┐ │
│ │ 02. 05. 2024    02. 05. 2024    07. 05. 2024│ │
│ │ ┌────────────┐ ┌────────────┐ ┌────────────┐ │ │
│ │ │ Satellite │ │ Satellite │ │ Satellite │ │ │
│ │ │  Image   │ │  Image   │ │  Image   │ │ │
│ │ └────────────┘ └────────────┘ └────────────┘ │ │
│ │ T33UUR-R022  T33UUQ-R022  T33UUR-R022   │ │
│ │ ☐       ☐       ☐          │ │
│ └──────────────────────────────────────────────────┘ │
│                                   │
│ Additional imagery rows below...               │
└──────────────────────────────────────────────────────────
┘
```

Key Features:

Date Range Selection:

- Automatically uses project's FROM_DATE to TO_DATE
- Shows all available satellite passes within this window

Product Selection:

- True color image (visible light)
- Other available products from dropdown

Image Status:

- **Usable** (blue eye icon) - Selected for analysis
- **Hidden** (crossed eye icon) - Excluded from calculations
- Use checkboxes to toggle which images to include

Image Information:

- Date of acquisition
- Grid reference (tile name, e.g., T33UUR-R022)
- Thumbnail preview
- Multiple images available for same area/date from different satellites

4.3 Understanding Satellite Image Coverage [info4agro]()

What the Map Shows:

- Detailed field-by-field boundaries (blue outlines)
- Individual field locations within project area
- Geographical context (rivers, roads, town names)
- Different image tiles may cover different field areas

Why Multiple Images?

- Different satellite passes (Sentinel-2A, Sentinel-2B, etc.)
- Different times of day
- Varying cloud coverage
- Different spectral/temporal resolution
- Allows selection of best quality images

Vegetation Index Processing:

Once images are selected, you can calculate:

Index	Use Case	Value Range
NDVI	General vegetation vigor	-1.0 to 1.0
EVI	Improved vegetation index	-1.0 to 1.0
SAVI	Soil-adjusted vegetation	-1.0 to 1.0
VCI	Vegetation condition	0 to 100
ALBEDO	Surface reflectance	0 to 1.0
NDSI	Snow/ice detection	-1.0 to 1.0

STEP 5: CREATING FERTILIZATION MAPS info4agro

Click **"Fertilisation"** to create Variable Rate Application (VRA) maps.

5.1 Prerequisites info4agro

Before creating fertilization maps, you must:

1. ✅ Download satellite imagery (Sat. images section)

2. ✅ Calculate at least one vegetation index (e.g., NDVI)

If Prerequisites Not Met:

text

```
┌──────────────────────────────────────────────────┐
│                          │
│          😟  No usable indices found          │
│                          │
│ Please calculate at least one vegetation index   │
│ for the selected project                    │
│                          │
│ [Open satellite imagery tool]               │
│                          │
└──────────────────────────────────────────────────┘
```

Action: Click "Open satellite imagery tool" to go back to Sat. images section and process data.

5.2 Creating Fertilization Maps [Once Data Available]

Workflow:

text

```
1. SELECT FIELDS
   └─ Click on fields on map (SHIFT for multiple)

2. CHOOSE VEGETATION INDEX
   └─ Select from: NDVI, EVI, SAVI, etc.

3. CONFIGURE ZONES
   └─ Set number of zones (3, 5, 10, etc.)

4. SELECT IMAGE DATE
   └─ Pick satellite image date within project range

5. GENERATE ZONES
   └─ Click "Get zones" to calculate map

6. REVIEW RESULTS
   ├─ Visual zoning map (field with color zones)
   ├─ Zone statistics table
   │  ├─ Zone number
   │  ├─ Zone area (hectares)
```

```
|   └─ Recommended fertilizer (kg/ha)
└─ Summary statistics

7. EXPORT
└─ Save as: GeoJSON, Shapefile, or vehicle format
```

Output Uses:

- Import into precision application equipment
- Variable rate applicators (VRA)
- GPS-guided tractors
- Site-specific management decisions
- Economic optimization

STEP 6: PLANNING WITH CLIMATE DATA – PLACE info4agro

Click **"Place"** under the PLANNING section to access historical climate data for location-based analysis.

6.1 Interface Overview [screenshot:10 & 11]

Map Display:

- Regional-scale view of entire Czech Republic
- Multiple visualization layers:
  - **Regular map** - Roads, towns, geography
  - **Climate layer** - Color gradient showing annual precipitation
    - Light blue = lower precipitation
    - Dark blue = higher precipitation
  - **Baseline map** - Geographic reference

Controls:

text
```
Season start: [Jun 1] [📅]
Year: [2023] (blue button)
View: [Week] [Year]
Transparency: [──────●──] (adjustable)
```

6.2 Selecting a Location info4agro

How to:

1. Click anywhere on the map
2. A red circle marker appears at clicked location
3. System: "Generating charts!"
4. Right panel populates with climate data

6.3 Climate Data Charts [info4agro]

Once location is selected and data loads, you'll see four comprehensive climate diagrams:

Chart 1: Clima Diagram

text

```
Precipitation (mm)   Temperature (°C)
100 | ▓▓ ▓▓ ▓▓      ────────────────────── 40°
 50 | ▓▓▓▓▓▓▓▓▓▓ ▓▓
  0 └──────────────────── Week 22-52       0°
    Week 22 24 26 ... 48 50 52

┌─ Blue bars: Total precipitation
└─ Red line: Average temperature
```

- Shows weekly patterns throughout year
- Identifies wet vs. dry periods
- Helps with irrigation planning
- Reveals frost risk timing

Chart 2: Water Balance

text

```
Water (mm)
80 |   ╱╲   ╱╲
60 |  ╱  ╲ ╱  ╲
40 | ╱    ╲╱    ╲
20 |╱         ╲
```

1. https://info4agro.com/agro-conditions
2. https://info4agro.com/info
3. https://www.youtube.com/watch?v=ZwnUev0fbKs&feature=youtu.be
4. https://info4agro.com/crop
5. https://info4agro.com/sensors
6. https://info4agro.com/sensors

7. https://info4agro.com/sim/manage
8. https://info4agro.com/sim/jobs
9. https://info4agro.com/place
10. https://info4agro.com/fertilisation