Synthetic Dataset Generation as a Service

Francesco Leotta

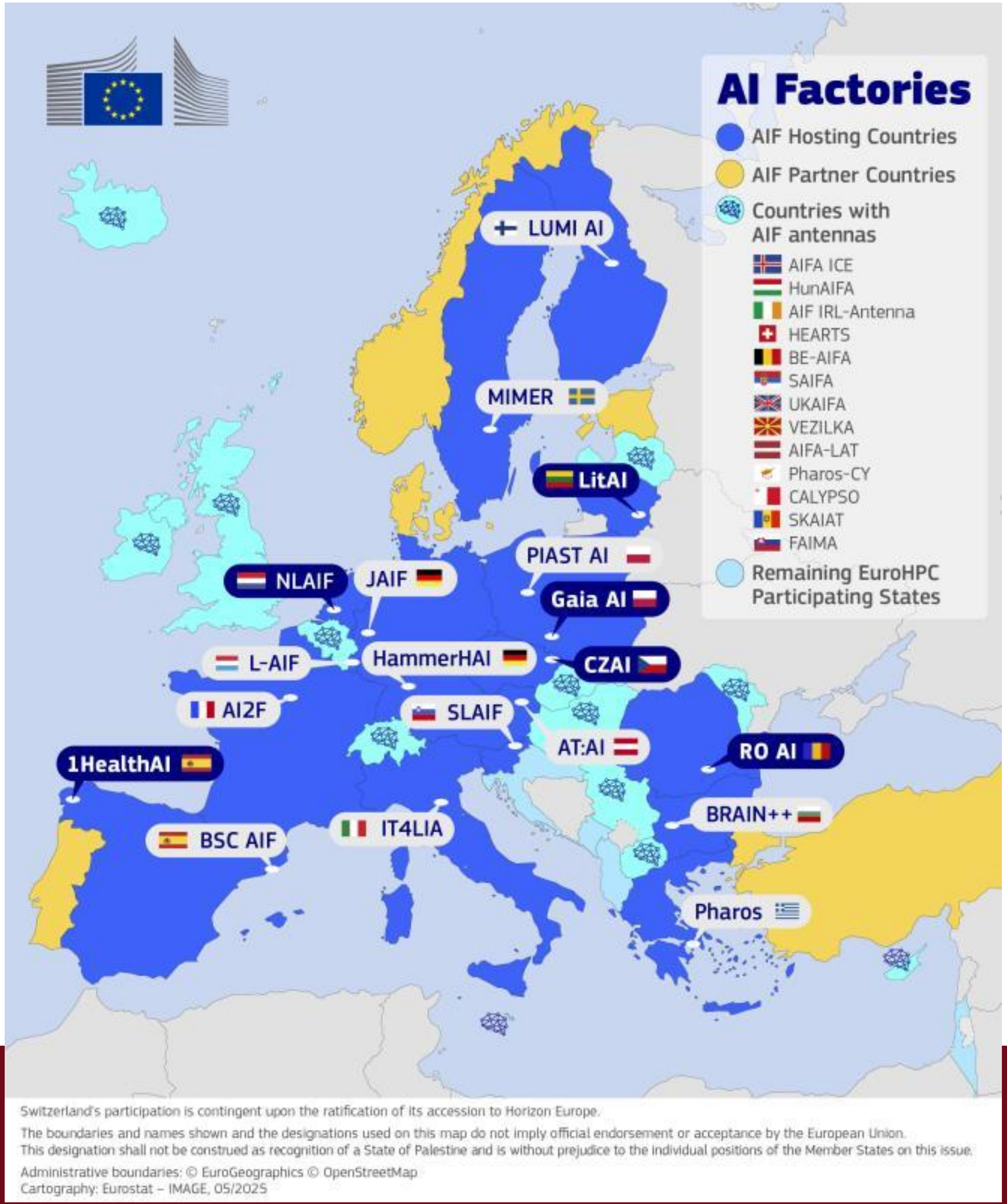
leotta@diag.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA, AUTOMATICA E GESTIONALE ANTONIO RUBERTI

Context: AI Factories

- An AI factory is the combination of:
  - An AI-Optimised Supercomputer
  - Associated Activities and Services
- Provide **computing power, data and support** to AI startups and researchers
- Support for the development of European AI models and applications



ITALIA AI FACTORY

Other Participating Member States

Federal Ministry
European and
International Affairs
Republic of Austria

REPUBLIC OF SLOVENIA
MINISTRY OF HIGHER EDUCATION, SCIENCE AND INNOVATION

Promoters

EuroHPC Joint Undertaking

AI Factories

MUR

ACN

Hosting Entity

CINECA

Regione Emilia-Romagna

INFN Istituto Nazionale di Fisica Nucleare

AI4I AI FOR INDUSTRY The Italian Institute of Artificial Intelligence

FBK FONDAZIONE BRUNO KESSLER

ItaliaMeteo

ICSC

iFAB INTERNATIONAL FOUNDATION BIG DATA AND ARTIFICIAL INTELLIGENCE FOR HUMAN DEVELOPMENT

SAPIENZA UNIVERSITÀ DI ROMA

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

UNIMORE UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

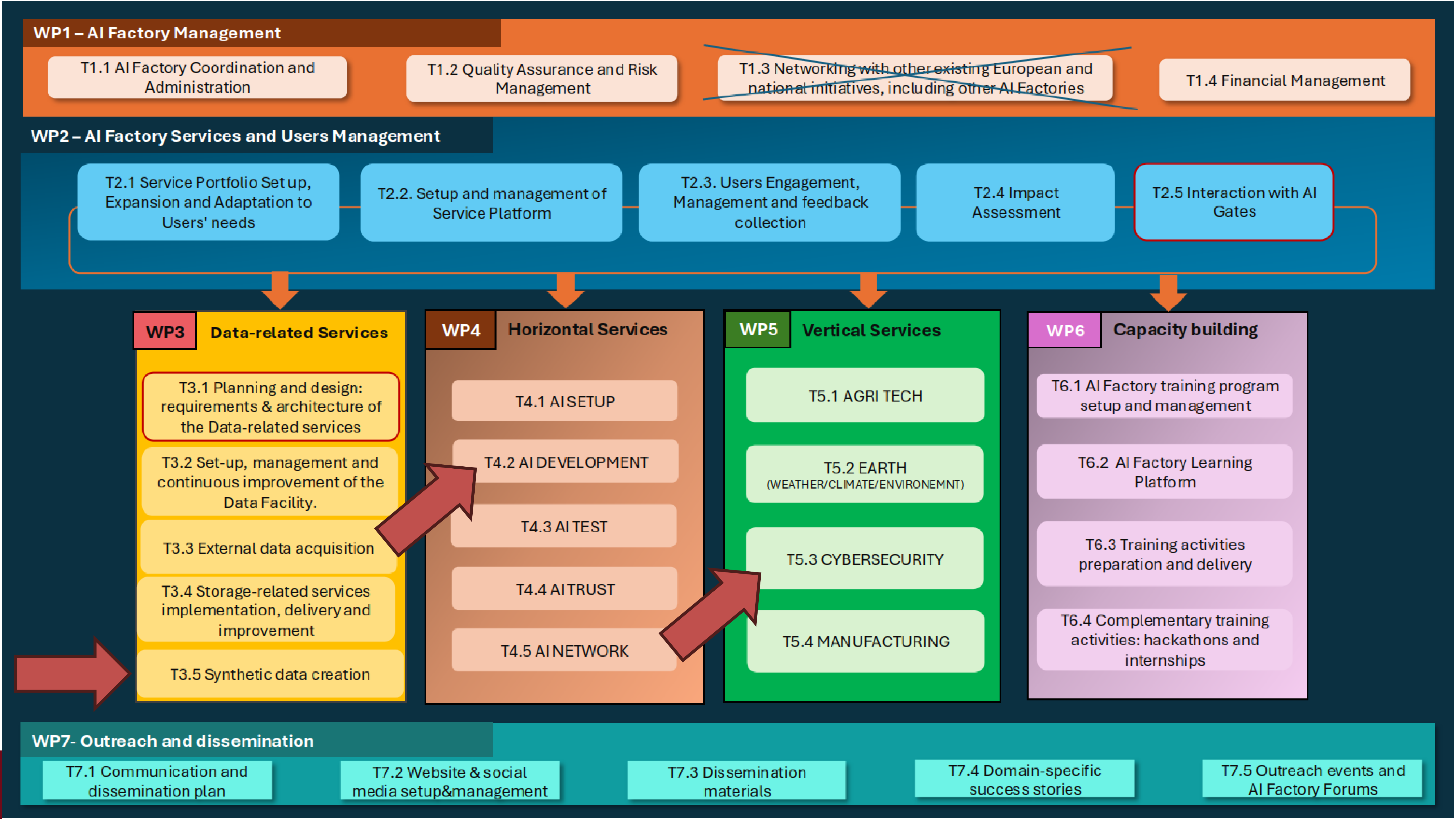
UNIVERSITÀ DI TORINO

ART-ER ATTRACTIVENESS RESEARCH TERRITORY

CONFINDUSTRIA

IT4LIA HPC Technological Infrastructure

- LEONARDO - With 3,500 processors and 14,000 state-of-the-art GPUs, it ranks among the world's top ten most powerful supercomputers
  - Combine HPC, AI, high-throughput computing and visualization applications.
- LISA - the new **AI-optimized partition within the LEONARDO system**.
- GAIA complements and **supports AI and HPC services through cloud technologies**.
- MEGARIDE - one of the most energy-efficient supercomputing systems in Europe, designed to handle the full range of AI use cases.



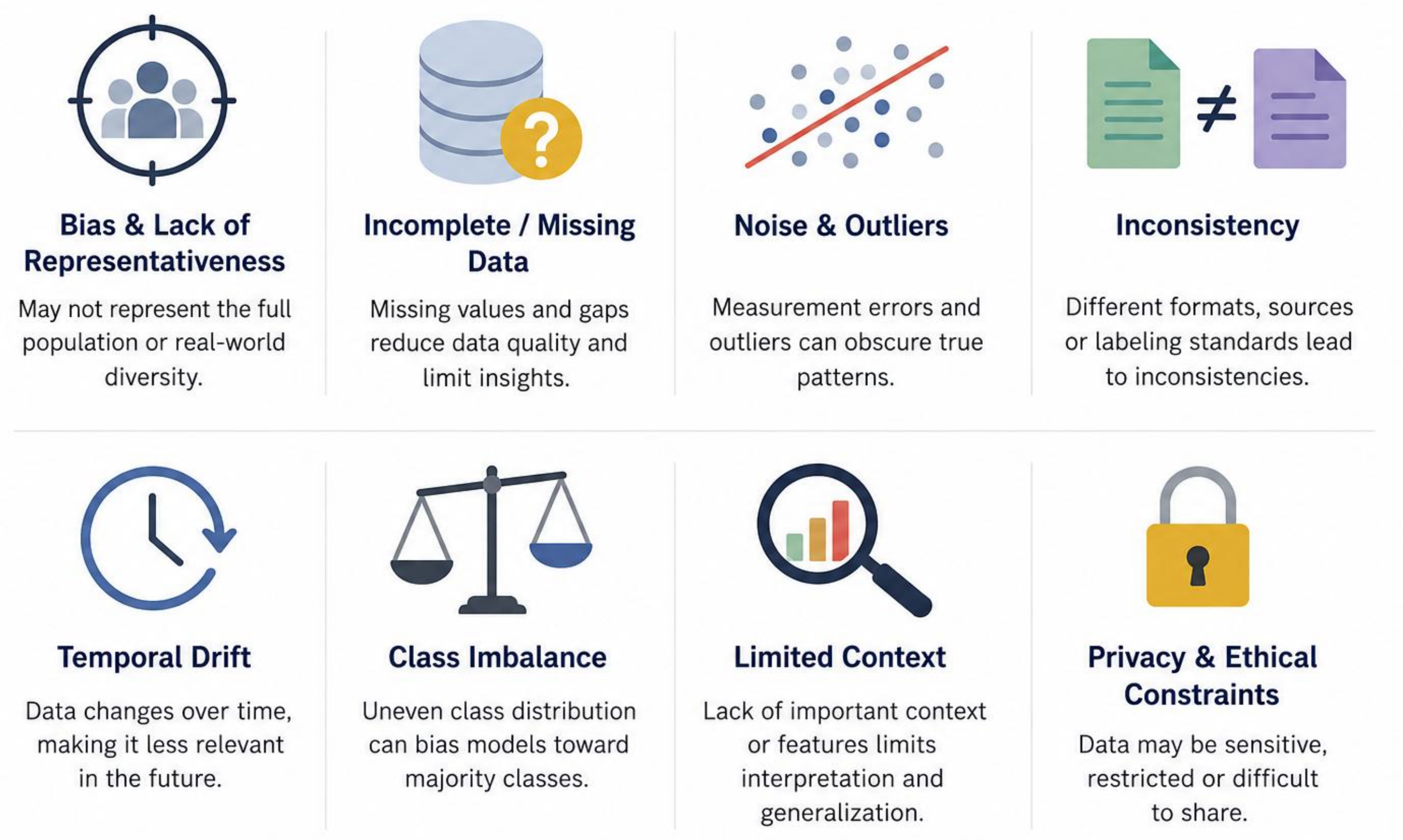
WP1 – AI Factory Management
T1.1 AI Factory Coordination and Administration
T1.2 Quality Assurance and Risk Management
~~T1.3 Networking with other existing European and national initiatives, including other AI Factories~~
T1.4 Financial Management
WP2 – AI Factory Services and Users Management
T2.1 Service Portfolio Set up, Expansion and Adaptation to Users' needs
T2.2. Setup and management of Service Platform
T2.3. Users Engagement, Management and feedback collection
T2.4 Impact Assessment
T2.5 Interaction with AI Gates
WP3
Data-related Services
T3.1 Planning and design: requirements & architecture of the Data-related services
T3.2 Set-up, management and continuous improvement of the Data Facility.
T3.3 External data acquisition
T3.4 Storage-related services implementation, delivery and improvement
T3.5 Synthetic data creation
WP4
Horizontal Services
T4.1 AI SETUP
T4.2 AI DEVELOPMENT
T4.3 AI TEST
T4.4 AI TRUST
T4.5 AI NETWORK
WP5
Vertical Services
T5.1 AGRI TECH
T5.2 EARTH
(WEATHER/CLIMATE/ENVIRONEMNT)
T5.3 CYBERSECURITY
T5.4 MANUFACTURING
WP6
Capacity building
T6.1 AI Factory training program setup and management
T6.2 AI Factory Learning Platform
T6.3 Training activities preparation and delivery
T6.4 Complementary training activities: hackathons and internships
WP7- Outreach and dissemination
T7.1 Communication and dissemination plan
T7.2 Website & social media setup&management
T7.3 Dissemination materials
T7.4 Domain-specific success stories
T7.5 Outreach events and AI Factory Forums

Why Synthetic Data Generation?

Limitations of real world-data are well discussed in literature

Synthetic data refers to artificially created data that mimics the statistical properties, structure, or features of real-world data

- Generated by algorithms rather than collected from real-world observations.
- Goals:
  - Training and fine tuning of new models
  - Validation of existing models
  - Simulation
- **Very often a big model to generate and a smaller model to be fine-tuned**



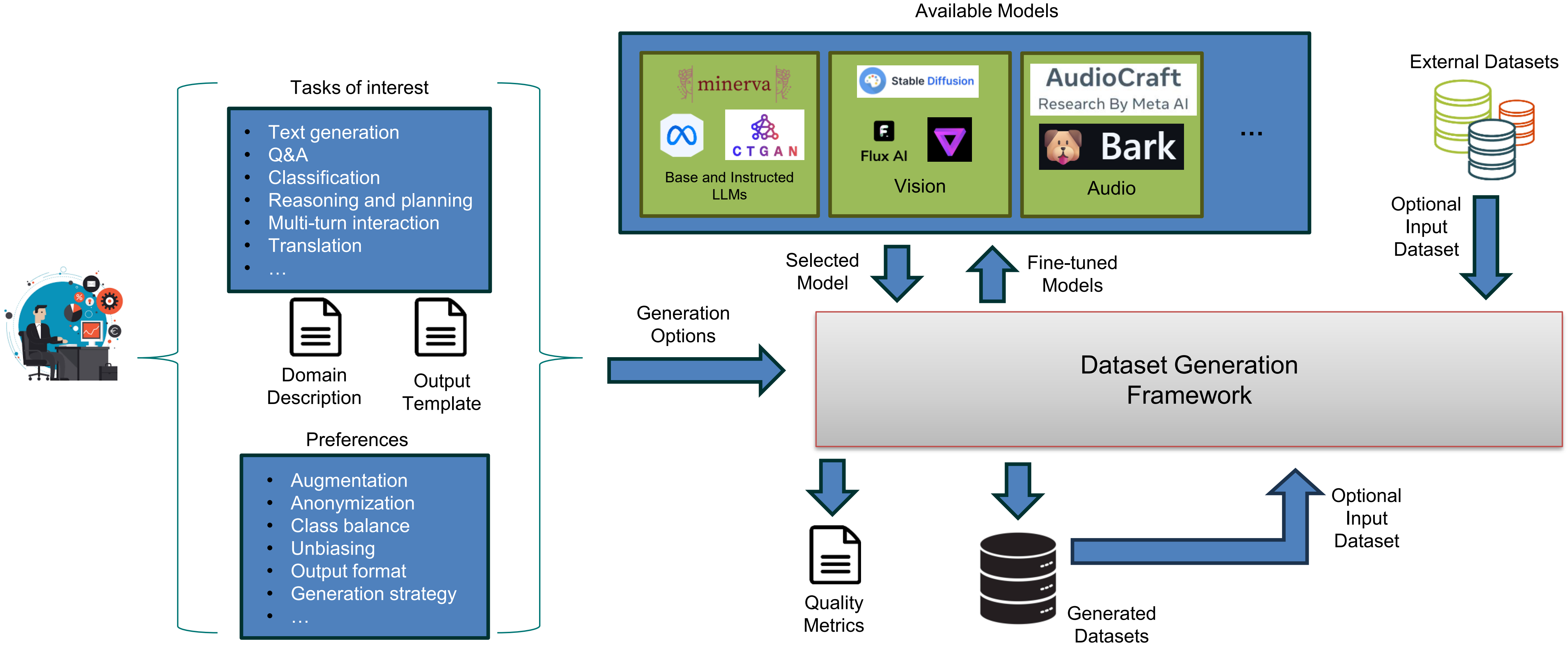
Generation vs Augmentation (1/2)

- Synthetic data generation and data augmentation are related, but conceptually and technically different
- **Data augmentation** involves modifying existing real data to create additional samples
- The goal is usually to increase dataset size, improve generalization, and make models robust to variations
- Key characteristics:
  - Works on real data that you already have
  - Changes are typically minor, controlled transformations that preserve the label
  - Often rule-based or simple stochastic transformations

Generation vs Augmentation (2/2)

- Examples of data augmentation:
  - Images: rotation, flipping, cropping, color jitter, adding noise. See https://github.com/aleju/imgaug
  - Text: synonym replacement, back-translation, slight paraphrasing. See makcedward/nlpaug: Data augmentation for NLP
  - Audio: pitch shift, time stretching, adding background noise.
  - Time-series: scaling, jittering, small shifts, window slicing.
- Pros:
  - Simple to implement.
  - Retains label fidelity.
  - Improves robustness of ML models.
- Cons:
  - Limited novelty; doesn't introduce fundamentally new patterns.
  - Can't fix dataset biases or create rare scenarios not in the original data.

IT4LIA – Dataset Generation Framework



Text Generation

- **Goal**: Create realistic sentences, documents, or structured text.
- **Approaches**:
  - Language Models (LMs): e.g., GPT, LLaMA, T5, etc. generate text token by token.
  - Template-based generation: Fill templates with different entities (common for structured text like invoices or resumes).
  - Controlled generation: Use prompts or conditioning to steer generation toward a topic, style, or structure.
- **Applications**:
  - Training NLP models without using sensitive text → Filtering and anonymization
  - Data augmentation for low-resource languages → Full text generation
  - Generating dialogues for chatbots → Q&A for instruct tuning
  - Generating tabular data (e.g., credit scoring)
  - Dataset conversion tasks (e.g., translation, classification)

Simple Example: LLM-Based Text Generation

Please generate a tabular dataset containing the following information: sex, income, country of provenance. In addition the dataset should countain a final column saying if the person is liable for a loan or not.

`llama-3-8b-lexi-uncensored`

Here's a sample dataset that meets your requirements:

Sex	Income	Country of Provenance	Loan Eligibility
Male	5000	USA	Yes
Female	3000	Canada	No
Male	8000	UK	Yes
Female	4000	Australia	No
Male	6000	Germany	Yes
Female	2000	France	No
Male	7000	Italy	Yes
Female	5000	Spain	No
Male	9000	Japan	Yes
Female	10000	China	Yes

Note: The income values are in USD and the country of provenance is just a random selection. The loan eligibility column is based on a simple rule that if the person's income is above a certain threshold (in this case, 6000), they are eligible for a loan. Otherwise, they are not.

- A simple example obtained using an uncensored version of LLAMA3
  - Uncensored models very important in this context to avoid guardrails
- The dataset is very biased
  - Sex → Loan Eligibility
  - Sex → Income
- **Generation is not that straightforward!!!**

Major Challenge: Fairness

- Direct requests to LLMs to generate data (especially tabular data) usually generates low quality results:
  - Very low fairness → Fairness metrics (see AI Fairness 360 repo)
    - **Bias in training data:** the model has learned from texts that reflect real-world practices and inequalities, including those present in the credit sector.
    - **Learned correlations:** even if you do not explicitly include race as a variable, other characteristics (for example geographic area, level of education, or occupation) can be correlated and act as proxies.
    - **Generation choices:** if the prompt is not very specific, the model may produce distributions that reflect statistical patterns observed in the data it learned from, including distorted ones.
- Additional challenges: e.g., problems with realism of certain attributes (e.g., official registration codes)

High Quality Text Generation Strategies - Naive

- **Prompt-based control:** Instead of simply asking «generate creditworthiness data”, specify constraints such as:
  - uniform distribution across groups
  - same default probability given income
  - no dependence between ethnicity and credit decision
  - explicitly defined distributions for each variable
- **Conditional generation:** You fix some variables and generate the rest. For example:
  - same income
  - same credit history
  - only demographic group varies
- **Post-processing:** After generation, analyze the dataset and correct imbalances:
  - oversampling underrepresented groups
  - undersampling overrepresented groups
  - reweighting samples
  - adjusting labels when appropriate

High Quality Text Generation Strategies - Advanced

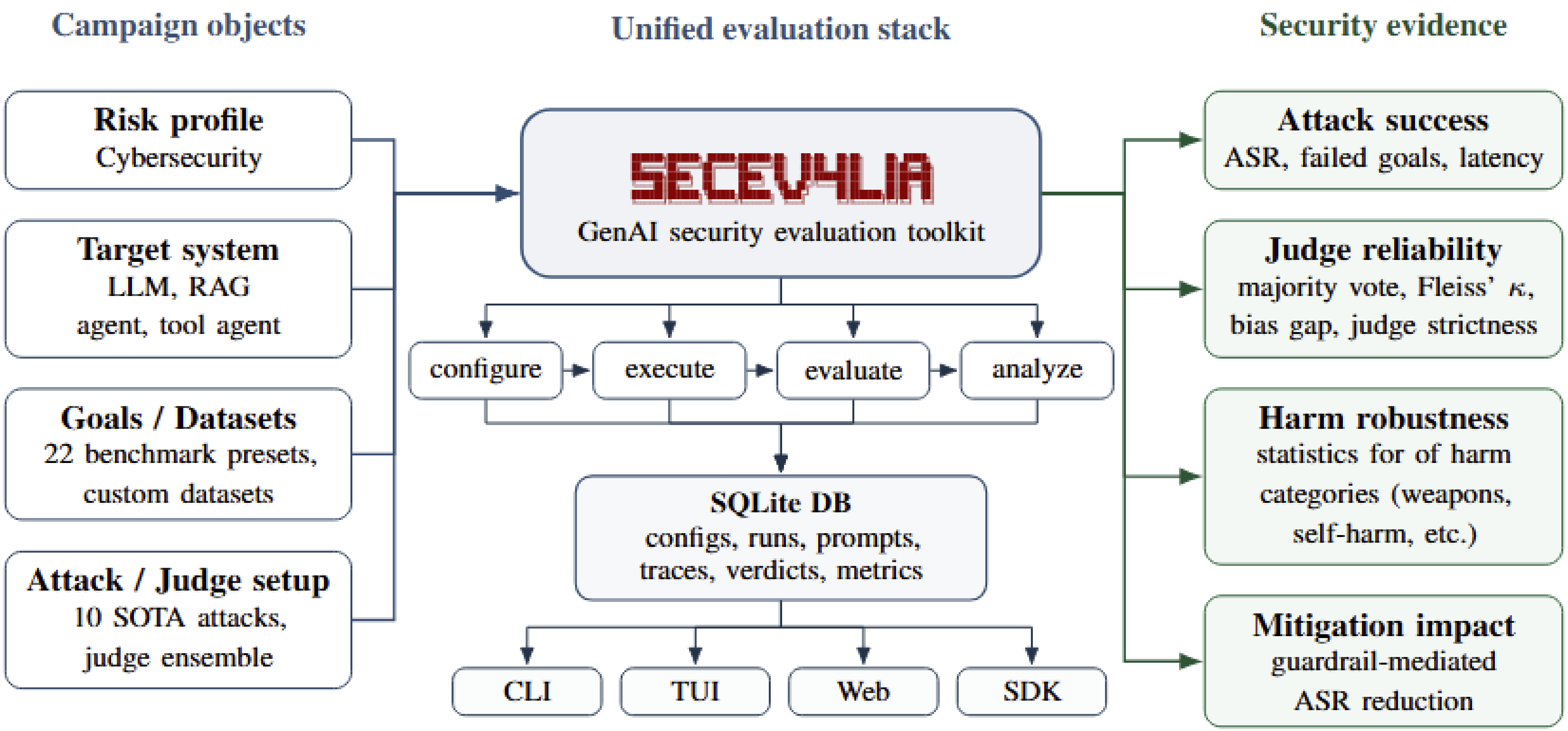
- **Iterative generation with auditing:**
  - generate a dataset
  - measure bias
  - identify causes
  - adjust prompts or parameters
  - regenerate
  - repeat until fairness metrics are within acceptable bounds
- **Use of specialized synthetic data models (see CTGAN)**

Dataset Based Image Generation: CTGAN

- CTGAN (Conditional Tabular GAN) is a type of generative adversarial network designed to create realistic **synthetic tabular data**
  - It learns patterns from real structured data and generates new, **statistically similar** rows while handling mixed data types and imbalanced categories
- Pros ☺:
  - generation is order of magnitude faster than LLMs
  - High quality data if the training dataset was high quality itself
- Cons ☹:
  - It requires a dataset
  - Low quality data if the training dataset was high quality itself → **Mixed generative approaches possible**

Application Scenario: SECEV4LIA

A red-teaming framework for structured, risk-driven evaluation of AI agents



Developing an Italian Judge

Prometheus 2 Approach
2405.01535

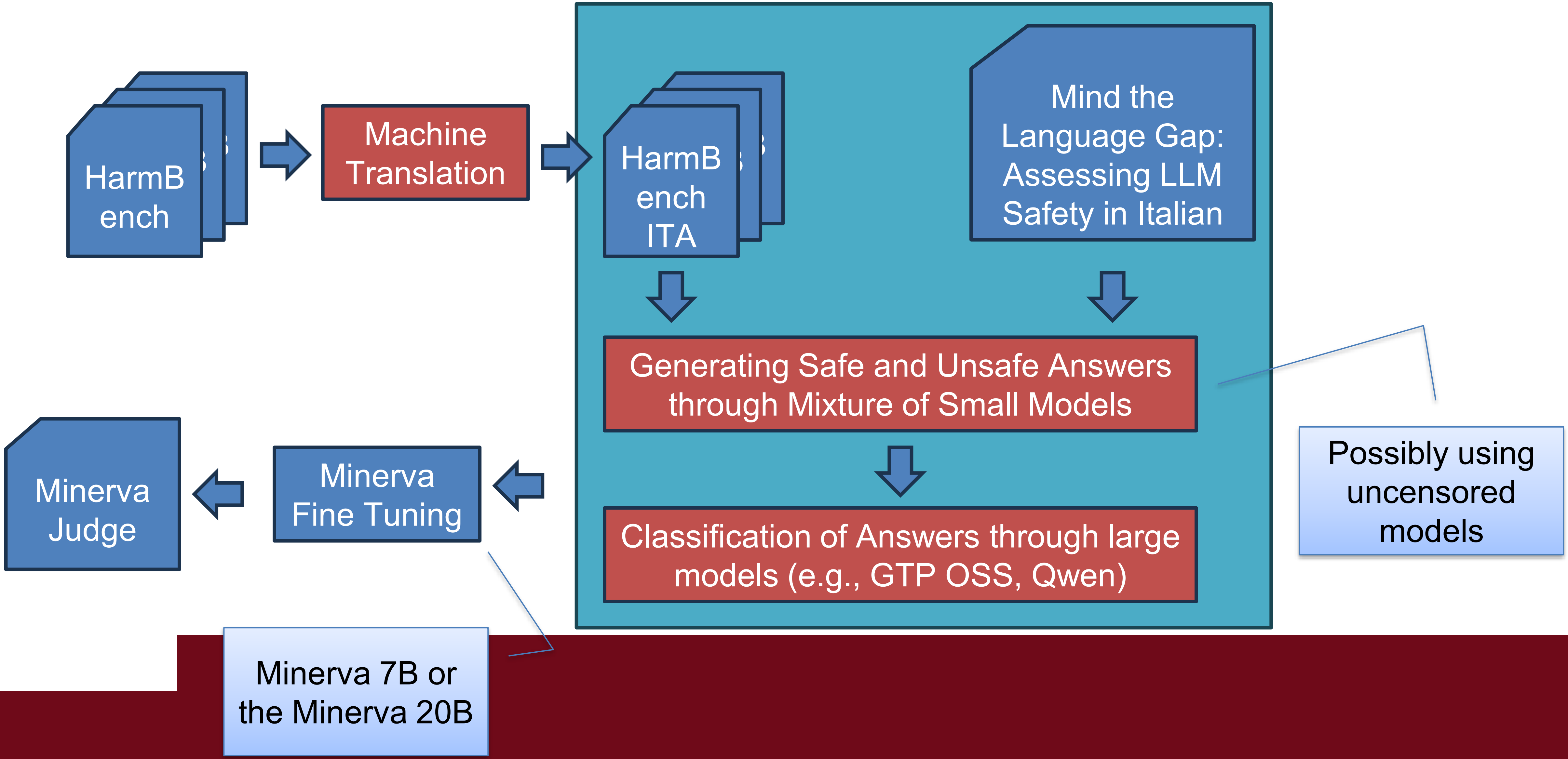


Image Generation

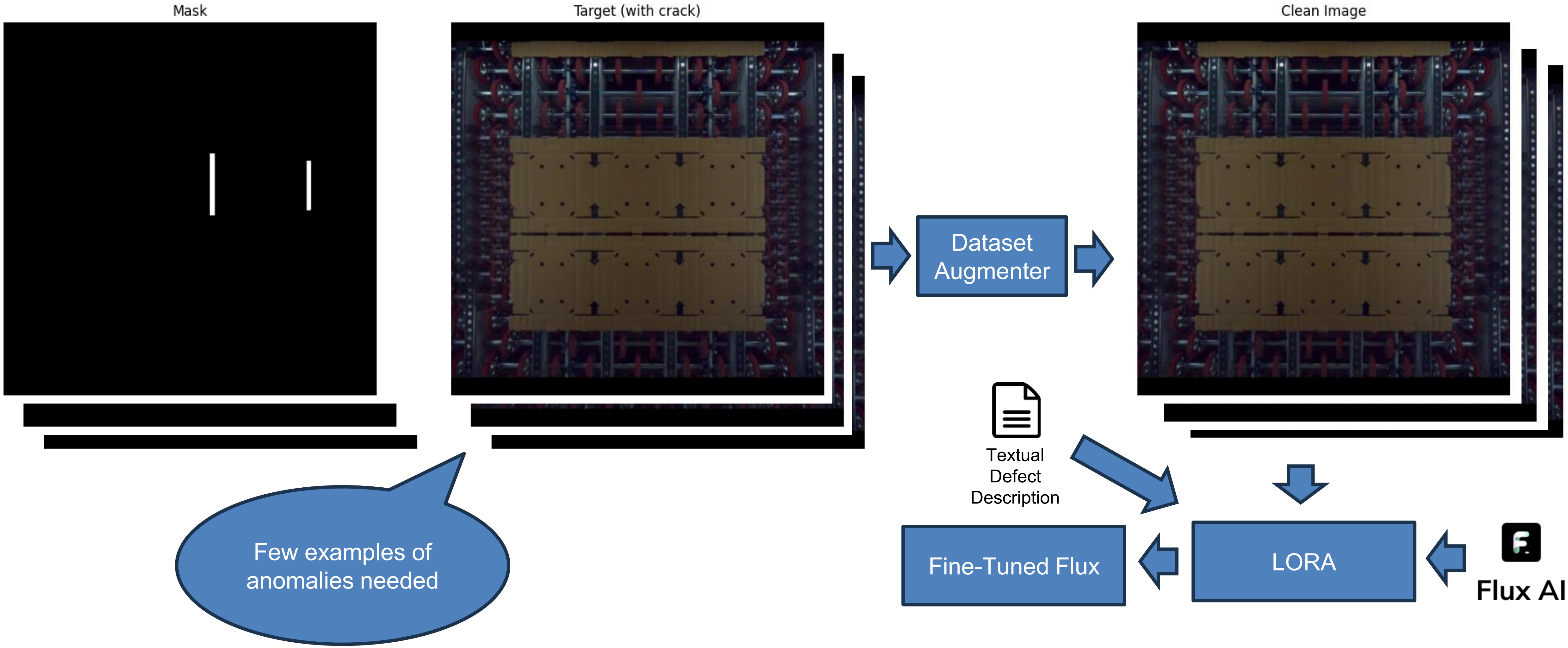
- **Goal**: Produce realistic images or entirely new visual scenes.
- **Approaches**:
  - Generative Adversarial Networks (GANs): Learn the distribution of real images and generate new ones (e.g., StyleGAN).
  - Diffusion Models: Start from noise and iteratively refine it to generate images (e.g., DALL·E, Stable Diffusion).
  - Variational Autoencoders (VAEs): Encode images into latent space and decode them to generate new images.
- **Applications**:
  - Augmenting datasets for computer vision tasks
  - Creative art and design
  - Virtual environments or synthetic training data for autonomous driving

Image Inpainting / Editing

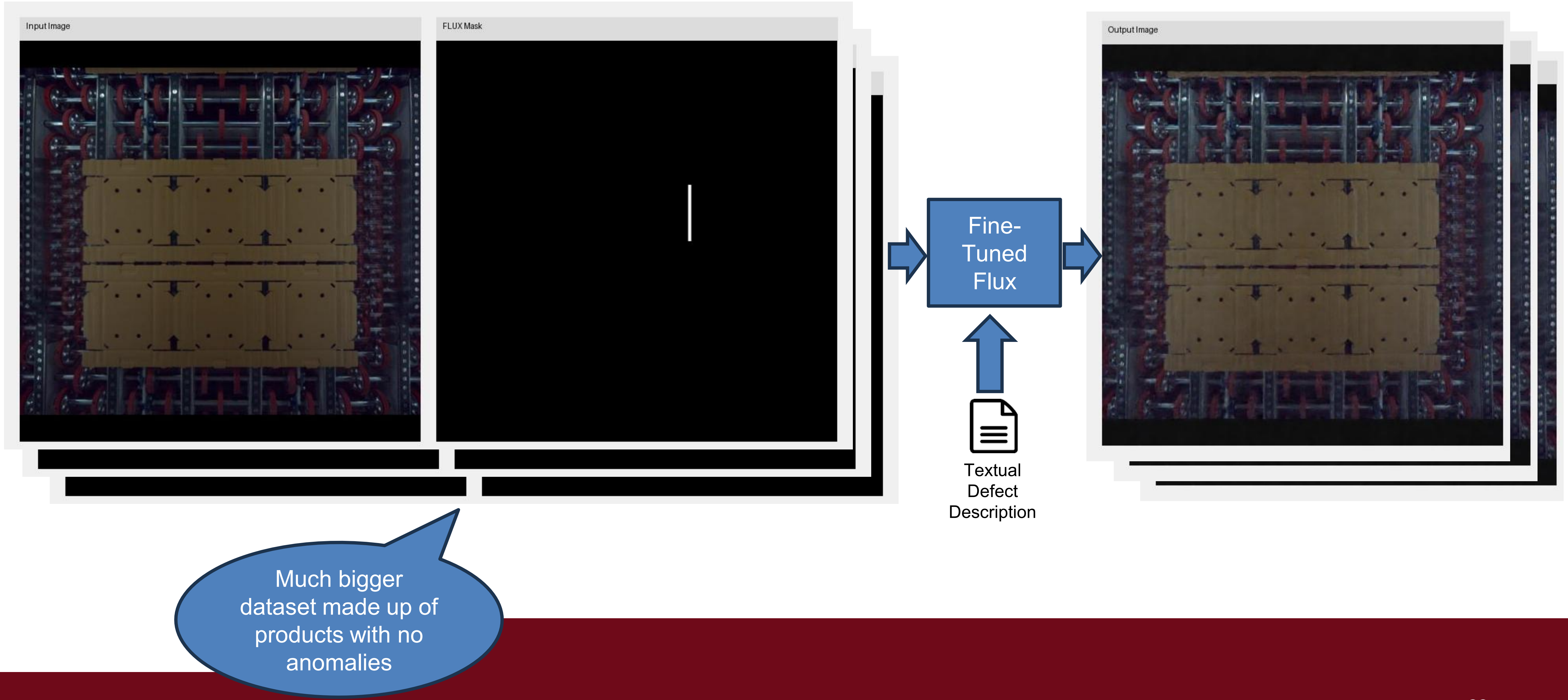
- **Goal**: Fill missing parts of an image or modify specific regions while maintaining realism.
- **Approaches**:
  - Mask-guided diffusion models: Predict pixels for masked areas conditioned on surrounding pixels.
  - GAN-based inpainting: Generate plausible content for occluded regions.
- **Applications**:
  - Removing unwanted objects from photos.
  - Restoring old images.
  - Data augmentation for computer vision tasks.

Industrial Case Study: Cardboard Quality Analysis

Industrial Anomaly Dataset Generation – Fine-Tuning



Industrial Anomaly Dataset Generation – Generation



Audio Generation

- **Goal**: Create realistic sound, speech, or music.
- **Approaches**:
  - Text-to-Speech (TTS) Models: Convert text to speech (e.g., Tacotron, WaveNet).
  - Music Generation Models: Learn patterns in audio to compose music (e.g., Jukebox by OpenAI).
  - Generative Diffusion for Audio: Generate waveforms or spectrograms, then convert back to audio.
- **Applications**:
  - Generating sound effects for virtual environments
  - **Generating sound effects for classification model training**
  - Music composition and remixing
  - Augmenting datasets for speech recognition or audio classification

Future features: Video Generation

- **Goal**: Create realistic video sequences, often with temporal consistency between frames.
- **Approaches**:
  - Frame-by-frame generation: Generate each frame individually using image generators, then enforce temporal smoothing.
  - Video GANs: Learn spatiotemporal patterns directly (e.g., MoCoGAN, TGAN).
  - Diffusion-based models: Extend diffusion techniques to video, predicting multiple frames conditioned on previous frames (e.g., Imagen Video, Make-A-Video).
  - Text-to-video models: Condition video generation on text prompts.
- **Applications**:
  - Creating synthetic training data for action recognition, tracking, or video understanding.
  - Entertainment: short clips, animations, or synthetic movie scenes.
  - Simulations for robotics or autonomous vehicles.

Future features: Time Series / Sensor Data

- **Goal**: Generate sequential data that simulates real-world sensor outputs or other temporal sequences.
- **Approaches**:
  - RNN / LSTM-based models: Learn temporal dependencies in sequences.
  - Temporal GANs: Generate realistic time-series data by learning sequential patterns.
  - Agent-based or physics-based simulation: Simulate sensor data from a modeled environment (common in robotics and IoT).
- **Applications**:
  - Autonomous vehicle sensor simulation (LiDAR, radar, cameras).
  - Financial data simulation for trading algorithms.
  - Healthcare: synthetic physiological signals like ECG or EEG.
  - IoT devices: synthetic sensor readings for testing and training AI models.

Future features: 3D Data / Point Clouds / Meshes

- **Goal**: Generate 3D shapes, environments, or objects for simulation or VR/AR applications.
- **Approaches**:
  - 3D GANs / VAE: Learn latent space for 3D voxel grids or point clouds.
  - Neural Radiance Fields (NeRFs): Represent 3D scenes as volumetric light fields for high-fidelity rendering.
  - Diffusion models for 3D: Generate point clouds or meshes via diffusion in latent space.
  - Procedural generation: Rule-based generation of 3D objects or terrains (common in games).
- **Applications**:
  - Autonomous driving: generating 3D LiDAR scenes.
  - Virtual reality / gaming: generating assets or full environments.
  - Robotics: simulated 3D environments for training.
  - Cultural heritage: reconstructing artifacts or buildings.

Future features: Multimodal Data

- **Goal**: Generate data that spans multiple modalities simultaneously, e.g., video with sound, images with text captions, or virtual avatars with speech and motion.
- **Approaches**:
  - Conditional generation: Condition one modality on another (text → image → video).
  - Joint latent models (e.g., VLMs): Encode multiple modalities into a shared latent space and generate coherently.
  - Diffusion-based multimodal models: Simultaneously model correlations between modalities (e.g., text-to-video, text-to-3D).
- **Applications**:
  - Text-to-video or text-to-3D models for content creation.
  - Virtual avatars with speech, gestures, and facial expressions.
  - Multimodal simulation environments for robotics.

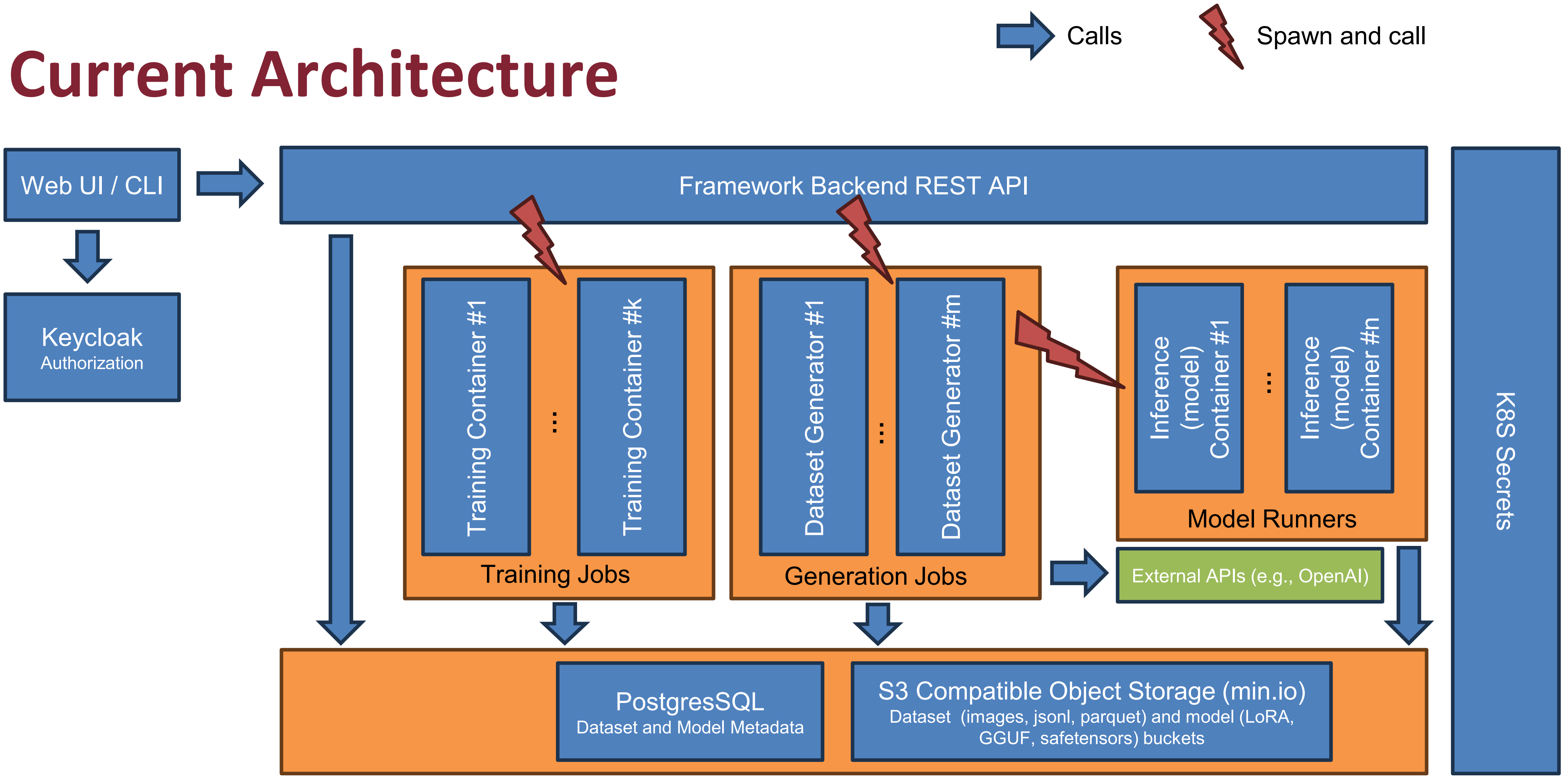
Why Servitizing Synthetic Dataset Generation?

1. It abstracts away a very complex pipeline
2. It standardizes quality and validation
3. It lowers the skill barrier dramatically
4. It improves collaboration between non-technical and technical users

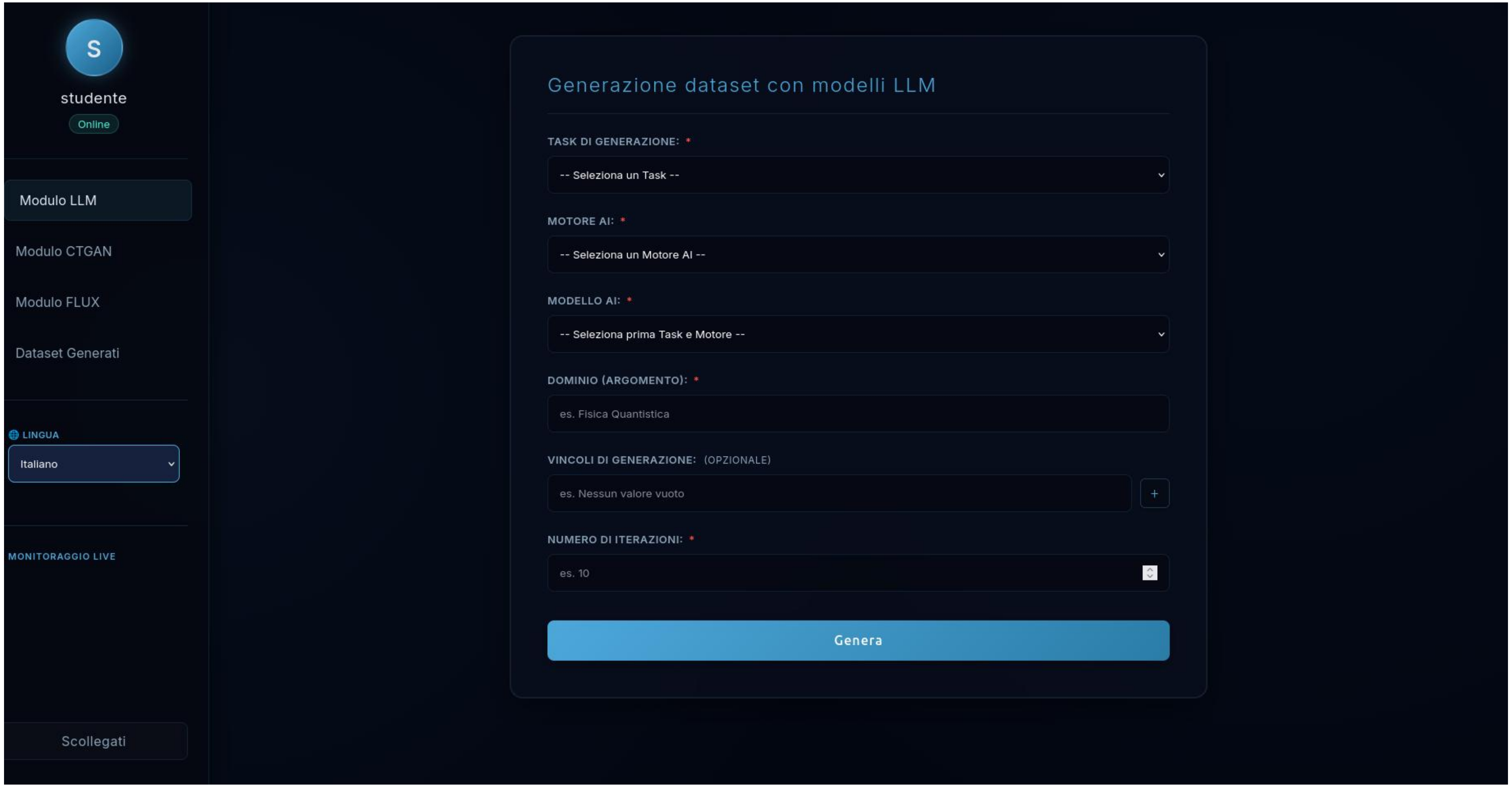
Service Fundamental Concepts

- **Jobs**
  - **Generation Jobs**: given a model, optional input data and generation parameters, it generates a certain amount of data
  - **Training Jobs**: given a training dataset and, optionally, a model it trains/fine-tune a new model
  - Both mapped to Kubernetes Jobs or to IaaC scripts
- **Collection of datasets**
  - Allows to generate datasets in successive steps
  - Contains a single dataset in the simplest case
- **Model**
- **Quality evaluation**
  - E.g., fairness through fairness 360

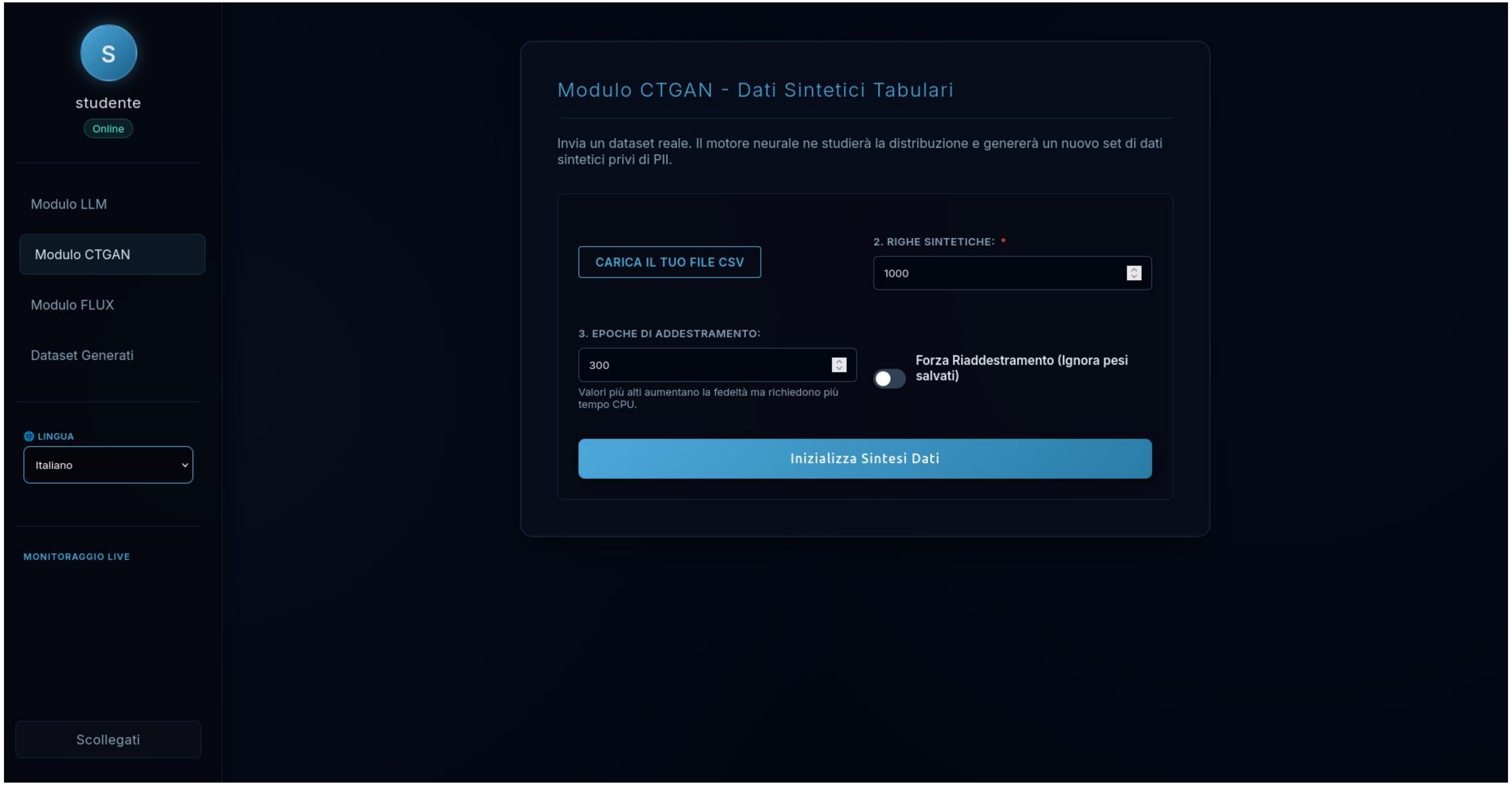
Current Architecture



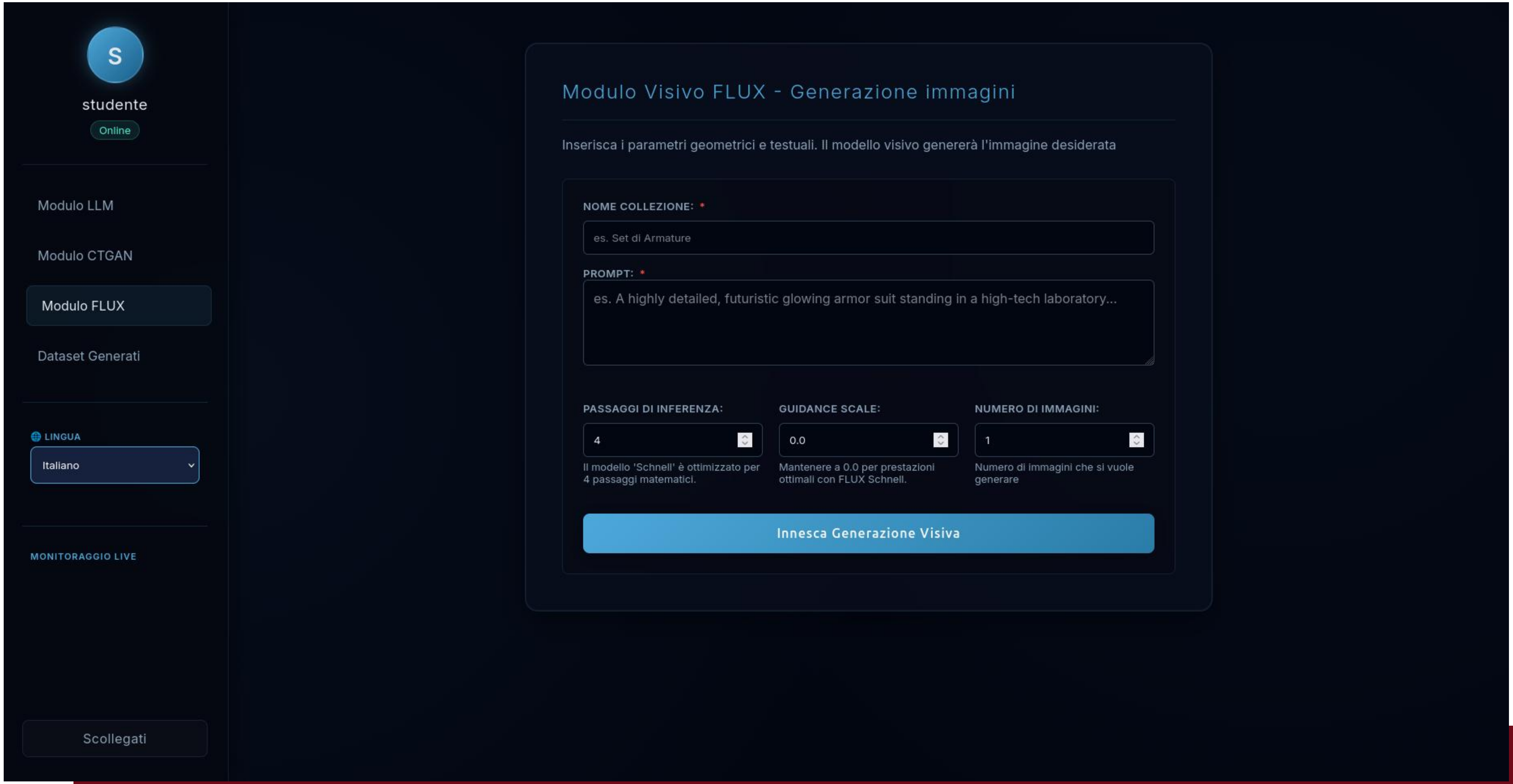
UI: Text Dataset Generation



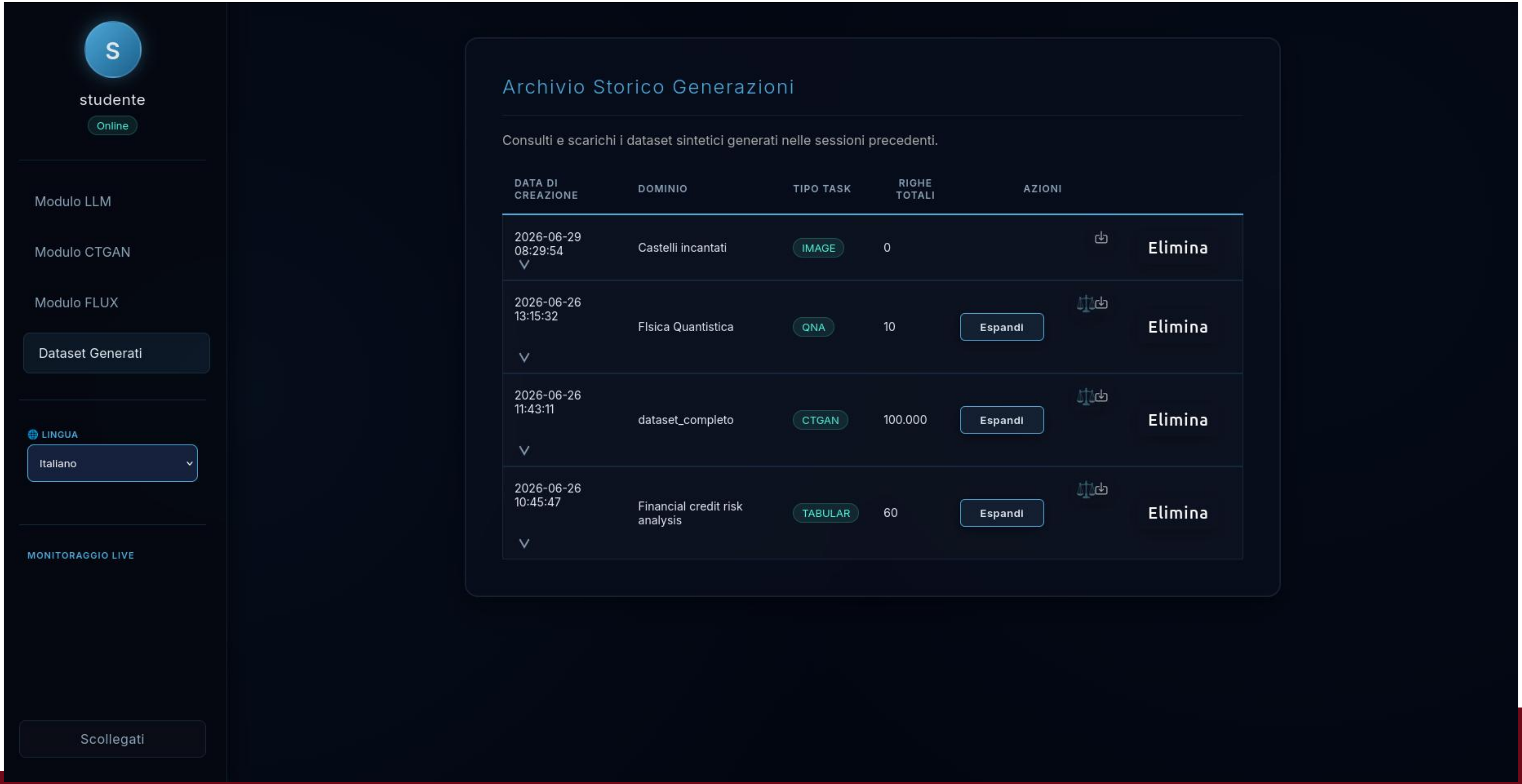
UI: Dataset based tabular data generation



UI: Prompt-based Image Generation



UI: Generation History



Research Challenges

- "Spec → Dataset" translation is still unsolved
  - No robust formalism for:
    - semantic specifications of datasets
    - automated decomposition into generative structure
- Ground truth realism is fundamentally unobservable
  - a multi-objective definition of realism that correlates with downstream utility
- Causal structure preservation
  - How do we generate datasets that preserve *causal invariants*, not just statistical ones?
- Compositionality and modular dataset generation
- Human-in-the-loop specification and correction