AI SPRINT
AI IN SECURE PRIVACY-PRESERVING COMPUTING CONTINUUM

# Design and Runtime Framework for Accelerating the Development of AI Applications in the Computing Continuum

Francesco Lattari, Politecnico di Milano

francesco.lattari@polimi.it

AI-SPRINT project has received funding from the European Union Horizoon 2020 research and innovation programme under Grant Agreement **No. 101016577.**

# AI-SPRINT Consortium

COORDINATOR

POLITECNICO MILANO 1863

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

GREMOIRE

Beck et al. work. together.

CLOUD & HEAT

7bulls.com

TTAnalysis

Trust-IT Services
communicating to markets

IDC

# AI-SPRINT Development Roadmap

State of the art analysis. Requirements analysis. Initial design of the architecture. (June 2021)

M6

M12

Initial Release and evaluation

(Dec. 2021)

Initial AI-SPRINT design and runtime tools integration (June 2022)

M18

M24

Second release and evaluation (Dec. 2022)

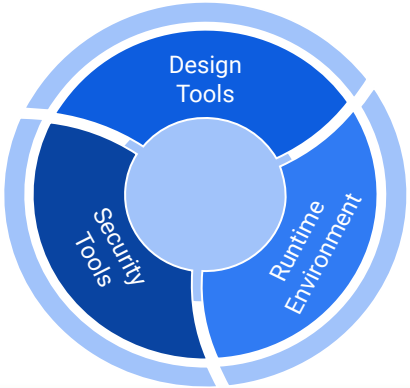
Final release and evaluation. Final AI-SPRINT design and runtime tools integration (June 2023)

Now

M30

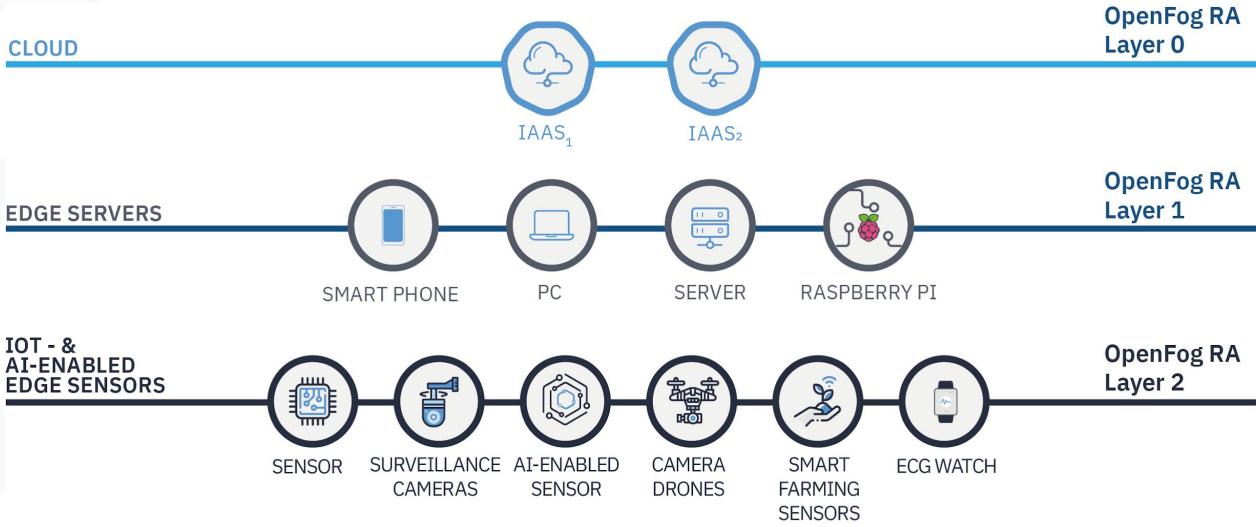
Evaluation & end of the project (December 2023)

M36



# Innovation for AI applications in Edge and Cloud Environments

- By 2026, AI worldwide market will approach $900 billion (CAGR 18.6%[1]) while edge computing will reach $324 billion (CAGR 13.6%[2])
- AI needs resources at the edge of the network
- New challenges from the infrastructural perspective

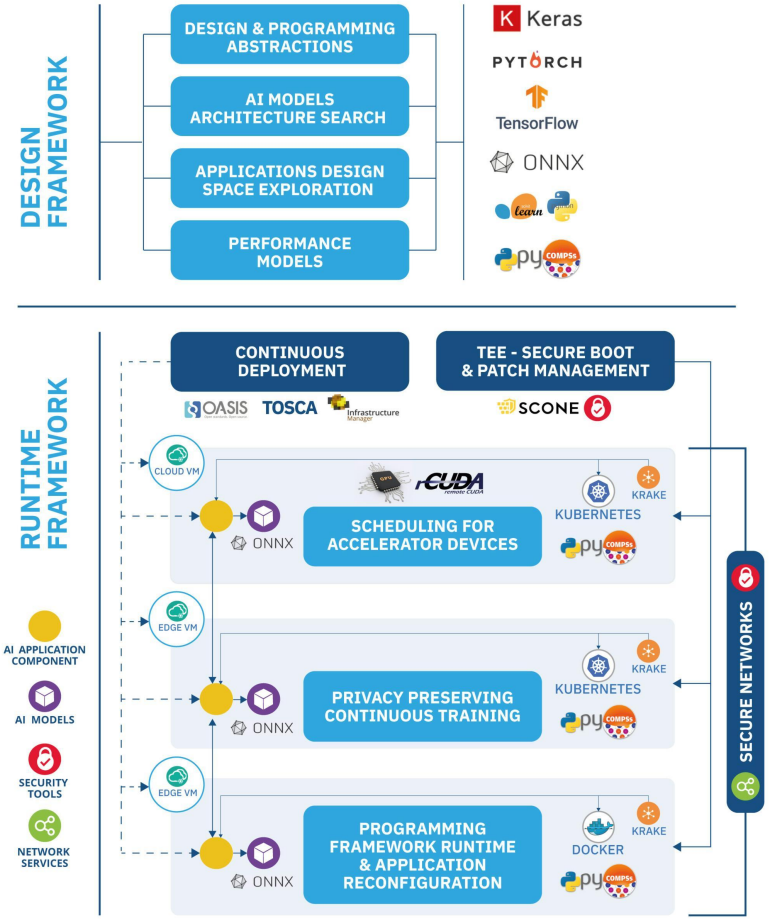



[1]IDC Semiannual Artificial Intelligence Tracker, July 2022

[2]IDC Worldwide Edge Spending Guide, August 2022

# AI-SPRINT objectives

- Simplified programming models
- Automated deployment and dynamic reconfiguration
- Secure execution of AI applications
- Highly specialized building blocks for privacy preservation, distributed training, and architecture enhancement
- Open source



# Main Technological Components

AI application definition through a template

Train high-quality models with QoS requirements

Resilient deployment of network functions

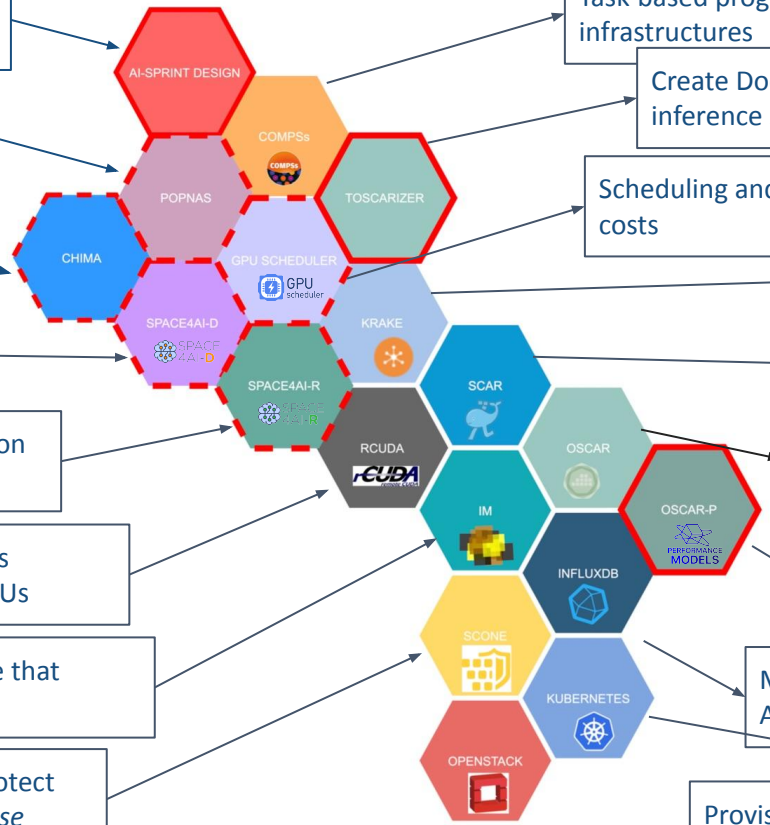
Design time component placement and resource selection decisions

Optimal components placement reallocation decision at runtime depending on the load

Remote GPUs as a Service that allows simultaneous multi-tenant use of GPUs

Infrastructure as Code (IaC) service that supports multiple Cloud back-ends

Confidential computing platform to protect data and code *at rest*, *in flight* and *in use*




Task-based programming model for distributed infrastructures

Create Docker images and TOSCA templates for inference services to deploy with IM and OSCAR

Scheduling and GPU allocation for energy/execution costs

Orchestrator engine to support migration of (GPU-based) training jobs

Serverless data-processing in AWS Lambda

Serverless Computing along the Continuum for event-driven data-processing workflows

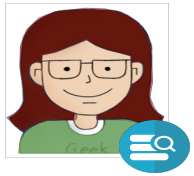
Profile an OSCAR app. on different hardware, train ML performance models from the collected data

Monitoring System: Infrastructure and Application

Container Orchestration Platform

Provision Virtual Machines (VMs)

# AI Application Design Workflow

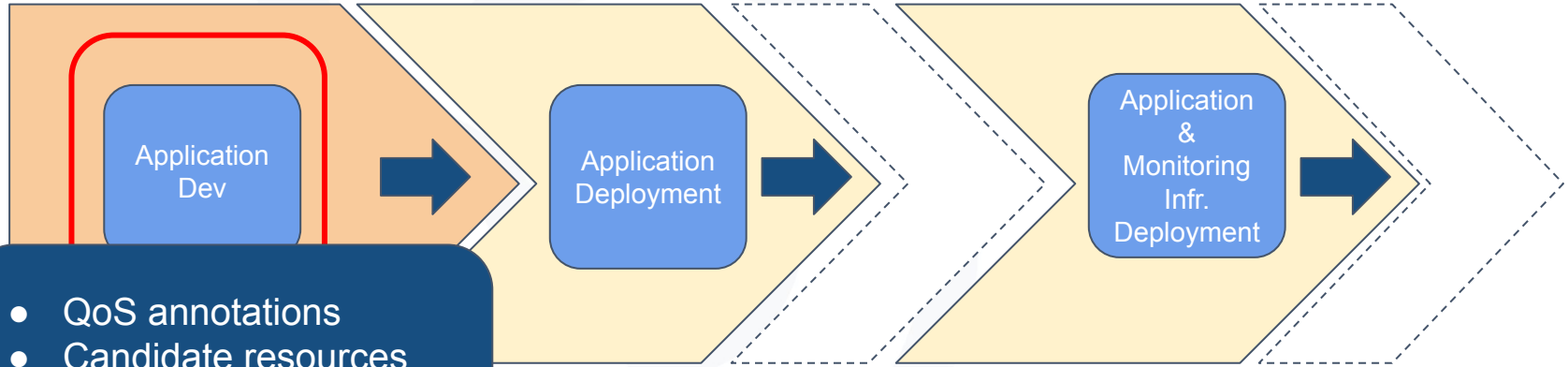


**Application Developer**

**Application Architect**

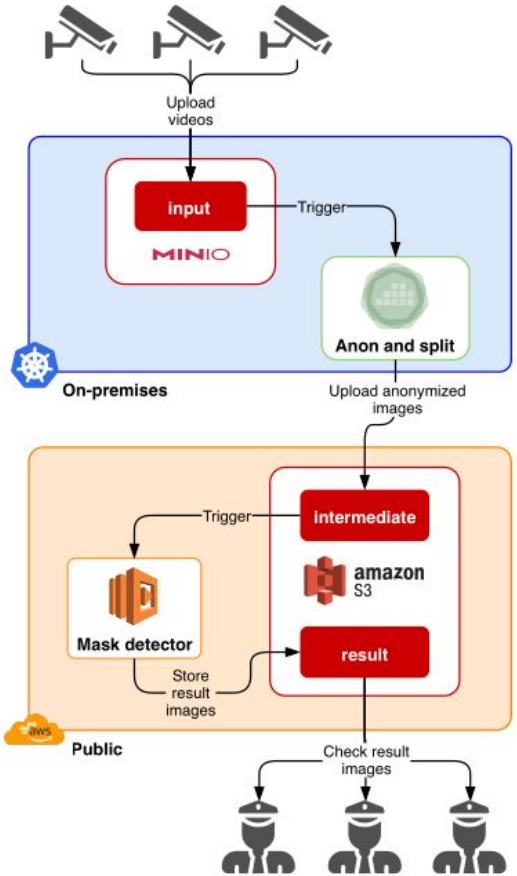
**Application Manager**

**Application Manager**

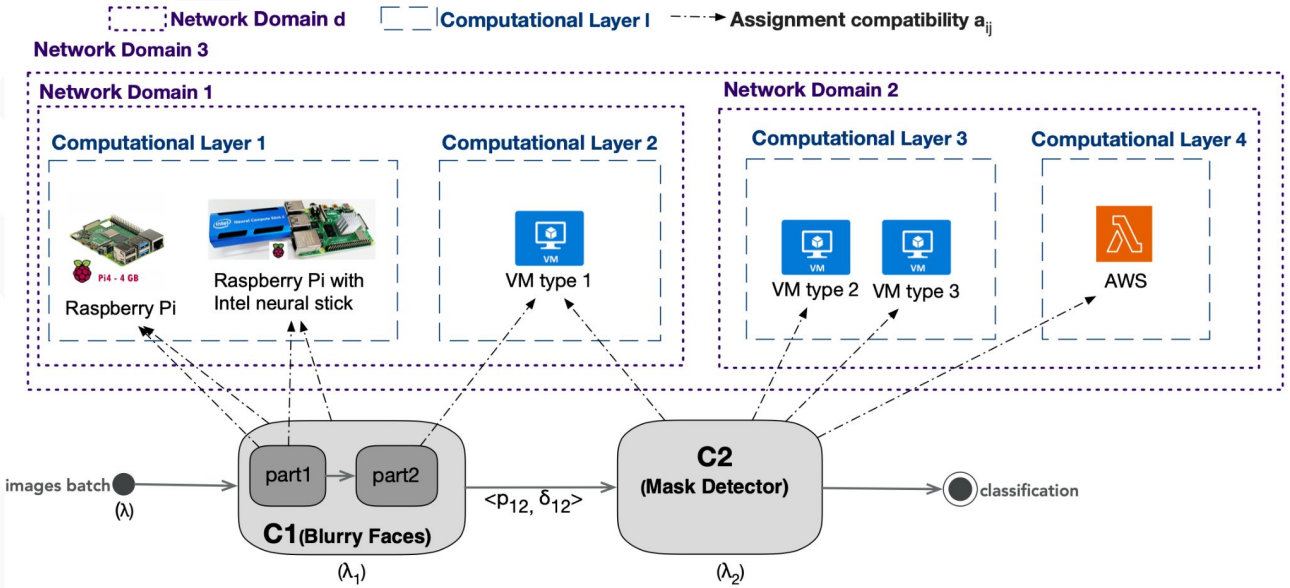



- QoS annotations
- Candidate resources and deployments
- DNNs partitioning

# AI-SPRINT Application




- Single or multiple Python applications, e.g., “Anon and split” and “Mask detector”
- Single or multiple candidate resources for each component
- Workflow execution orchestrated by **OSCAR** (https://oscar.grycap.net)



# AI-SPRINT Application Development

**Application Developer**

Annotated Python code

$C_1$ → $C_2$

DAG

**Application Manager**

Candidate Resources

**Application Architect**

Candidate Deployments

```
AI-SPRINT_MASK_DETECTION_APP
  aisprint
    deployments
    designs
    logs
  ams
  common_config
    application_dag.yaml
    candidate_deployments.yaml
    candidate_resources.yaml
  im
  oscar
  pycompss
  space4ai-d
  src
    blurry-faces-onnx
      onnx
        version-RFB-640.onnx
      main.py
      requirements.txt
      utils.py
    mask-detector
      cfg
        obj.names
        yolov3-tiny_obj_test.cfg
        yolov3-tiny_obj_train_tiny8.weights
        yolov3-tiny_obj_train.cfg
      main.py
      requirements.txt
```

```python
@component_name(name='mask-detector')
@exec_time(local_time_thr=10)
@device_constraints(ram=1024, vram=2048)
@security(trustedExecution=False, networkShield=False, filesystemShield=False)
def main(args):
```

```
System:
  name: ai-sprint_mask_detection_app
  components: ['blurry-faces-onnx', 'mask-detector']
  dependencies: [['blurry-faces-onnx', 'mask-detector', 1]]
```

```
System:
  name: Mask Detection Application
  NetworkDomains:
    ND1:
      name: Network Domain 1
      AccessDelay: 0.00000277
      Bandwidth: 40000
      subNetworkDomains: []
      ComputationalLayers:
        computationalLayer1:
          name: Edge Layer
          number: 1
          type: PhysicalAlreadyProvisione
          Resources:
            resource1:
              name: RaspPi
              totalNodes: 3
              description: Raspberry PI
              cost: 0.6
              memorySize: 4096
              operatingSystemDistributi
              operatingSystemType: Linu
              operatingSystemVersion: 1
              operatingSystemImageId: N
```

```
Components:
    component1:
      name: blurry-faces-onnx
      candidateExecutionLayers: [1,2]
      Containers:
        container1:
          image: registry.gitlab.polimi.it/ai-s
          memorySize: 2048
          computingUnits: 0.9
          trustedExecution: False
          networkProtection: False
          fileSystemProtection: False
          GPURequirement: False
          candidateExecutionResources: [RaspPi]
        container2:
          image: registry.gitlab.polimi.it/ai-s
          memorySize: 2048
          computingUnits: 0.9
          trustedExecution: False
          networkProtection: False
          fileSystemProtection: False
          GPURequirement: False
          candidateExecutionResources: [VM1]
```

# AI-SPRINT Quality of Service (QoS) Annotations

| Annotation name |
| --- |
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| *@aisprint.early_exits_model* |
| *@aisprint.model_performance* |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

# AI-SPRINT Quality of Service (QoS) Annotations

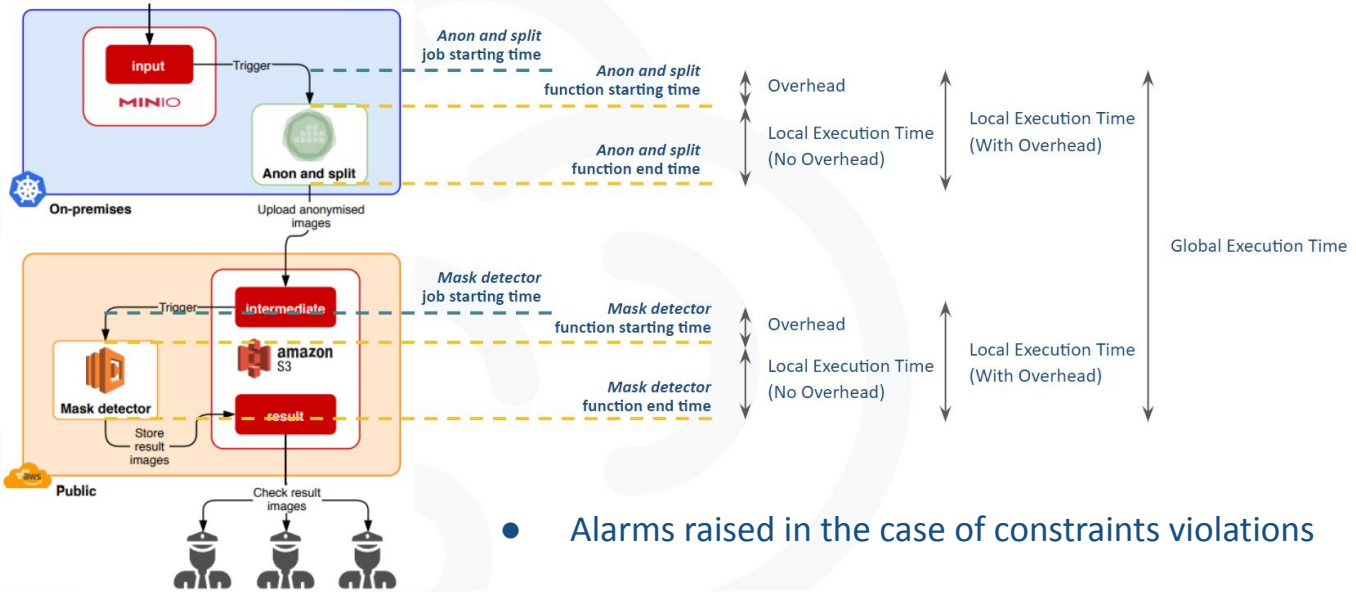
| Annotation name |
|---|
| *@aisprint.component_name* |
| ***@aisprint.exec_time*** |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| *@aisprint.early_exits_model* |
| *@aisprint.model_performance* |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

```
def exec_time(local_time_thr, global_time_thr, prev_components)
```

Allows users defining execution time constraints for single (local) or multiple components (global)

- Constraints are automatically monitored by the **AI-SPRINT Monitoring Subsystem (AMS)**

https://gitlab.polimi.it/ai-sprint/monitoring-subsystem




- Alarms raised in the case of constraints violations

# AI-SPRINT Quality of Service (QoS) Annotations

| Annotation name |
| --- |
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| ***@aisprint.expected_throughput*** |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| *@aisprint.early_exits_model* |
| *@aisprint.model_performance* |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

```
def expected_throughput(rate)
```

Allows the users to define the expected application throughput

- i.e., the expected invocation rate expressed as the number of invocations per time unit (number of invocations per second)

# AI-SPRINT Quality of Service (QoS) Annotations

| Annotation name |
|---|
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| ***@aisprint.partitionable_model*** |
| *@aisprint.device_constraints* |
| *@aisprint.early_exits_model* |
| *@aisprint.model_performance* |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

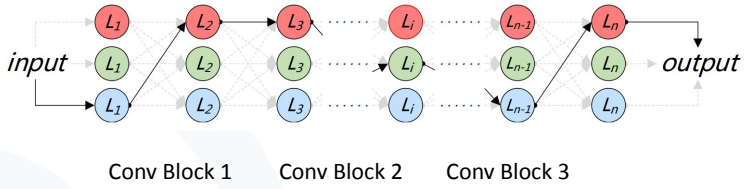
```
def partitionable_model(onnx_file, num_partitions)
```

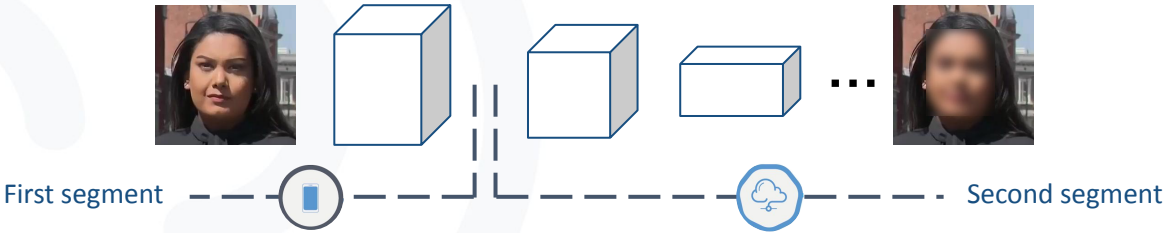
Gives the user the possibility of defining a *partitionable* Deep Neural Network (DNN), provided using the **Open Neural Network Exchange (ONNX)** format

- i.e., the model is divided in different parts, which can be executed on different computational layers (L)

ONNX RUNTIME

https://onnxruntime.ai/




- Automatic split performed by the SPACE4AI-D-Partitioner tool (https://gitlab.polimi.it/ai-sprint/ai-sprint-design/-/tree/master/src/aisprint/space4aidpartitioner)

# AI-SPRINT Quality of Service (QoS) Annotations

| Annotation name |
| --- |
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| ***@aisprint.device_constraints*** |
| *@aisprint.early_exits_model* |
| *@aisprint.model_performance* |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

```
def device_constraints(ram, vram)
```

The annotation allows the users to specify a set of minimal resources that the device on which the component will be deployed must have. In particular the

- ***ram***: minimum memory in GB required to run the annotated component
- ***vram***: minimum video memory in GB required to run the annotated component. A *vram > 0* implicitly highlights the need for a Graphics Processing Unit (GPU).

# AI-SPRINT Quality of Service (QoS) Annotations

```
def early_exits_model(onnx_file, condition_function, transition_probabilities)
```

| Annotation name |
|---|
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| ***@aisprint.early_exits_model*** |
| *@aisprint.model_performance* |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

Similarly to the partitionable models, allows automatic partitioning of DNN with early exits

- Execution can be stopped earlier in the network based on user-defined conditions. E.g., BranchyNet: Fast Inference via Early Exiting from Deep Neural Networks

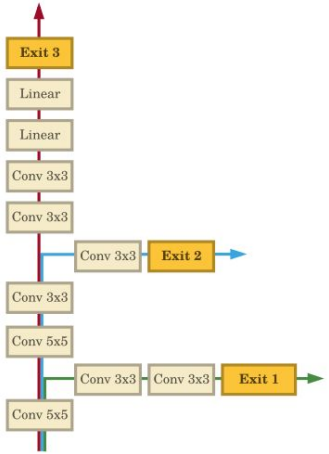
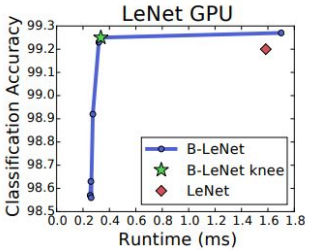
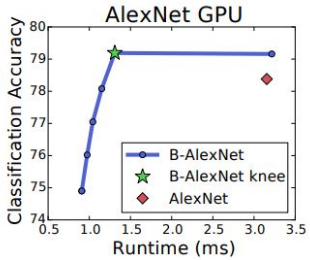
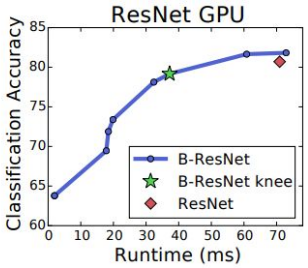



Fig. 1: A simple BranchyNet with two branches added to the baseline (original) AlexNet. The first branch has two convolutional layers and the second branch has 1 convolutional layer. The "Exit" boxes denote the various exit points of BranchyNet.



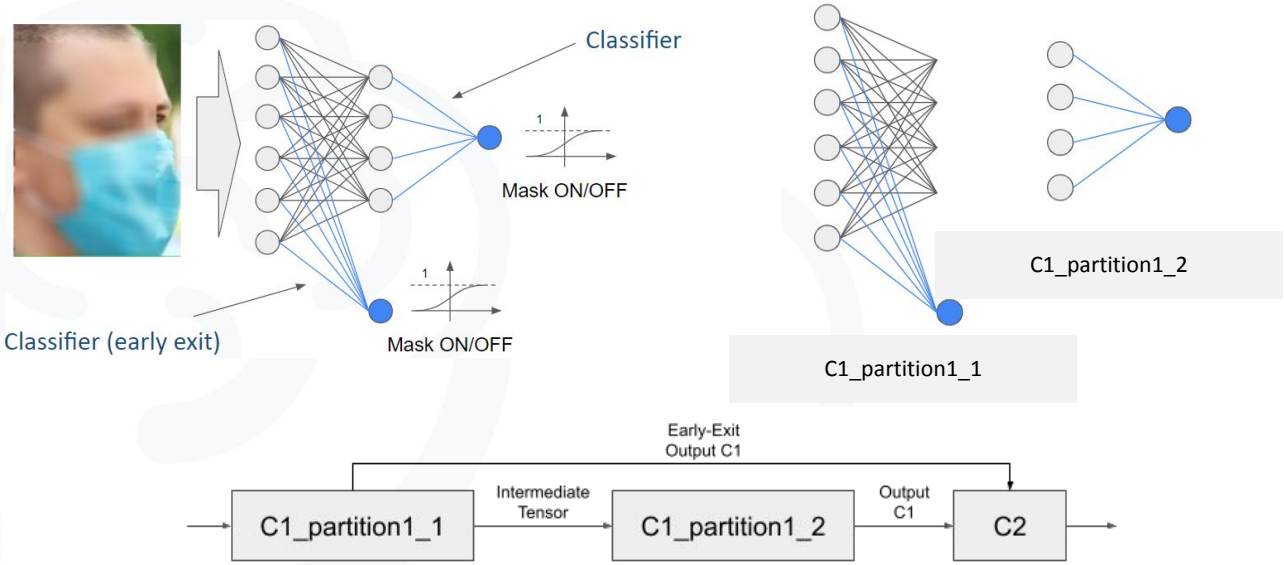
# AI-SPRINT Quality of Service (QoS) Annotations

```
def early_exits_model(onnx_file, condition_function, transition_probabilities)
```

| Annotation name |
|---|
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| ***@aisprint.early_exits_model*** |
| *@aisprint.model_performance* |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

Similarly to the partitionable models, allows automatic partitioning of DNN with early exits

- AI-SPRINT automatically split the DNN-based components at early exits, generating new components corresponding to the network segments



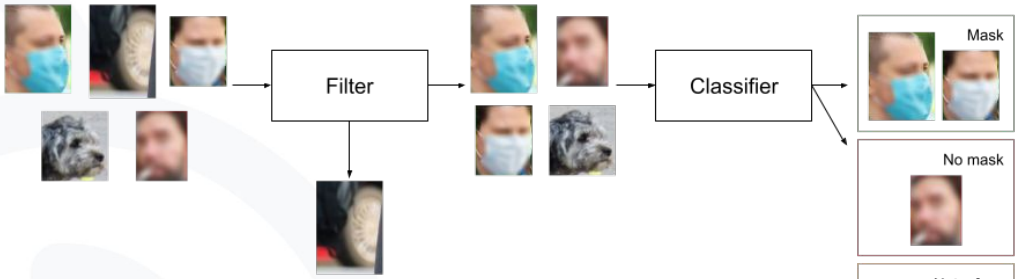
# AI-SPRINT Quality of Service (QoS) Annotations

| Annotation name |
|---|
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| *@aisprint.early_exits_model* |
| ***@aisprint.model_performance*** |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

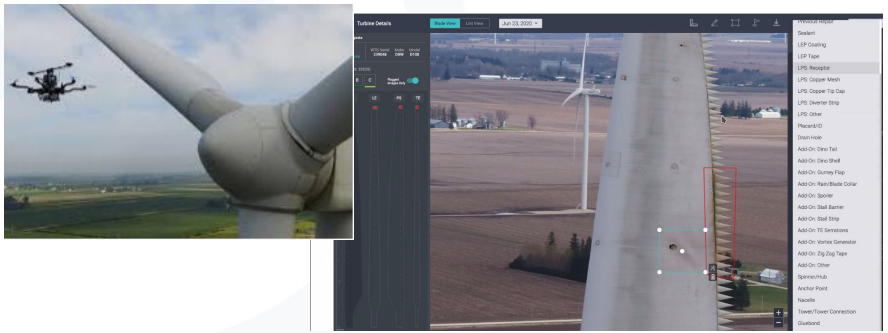
```
def model_performance(metric, metric_thr,filtered_class)
```

Allows defining applications with degraded performance

- Use case: filter+classifier applications




- E.g., maintenance and inspection (windmill blade damage)



# AI-SPRINT Quality of Service (QoS) Annotations

| Annotation name |
|---|
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| *@aisprint.early_exits_model* |
| ***@aisprint.model_performance*** |
| *@aisprint.detect_metric_drift* |
| *@aisprint.security* |

```
def model_performance(metric, metric_thr,filtered_class)
```

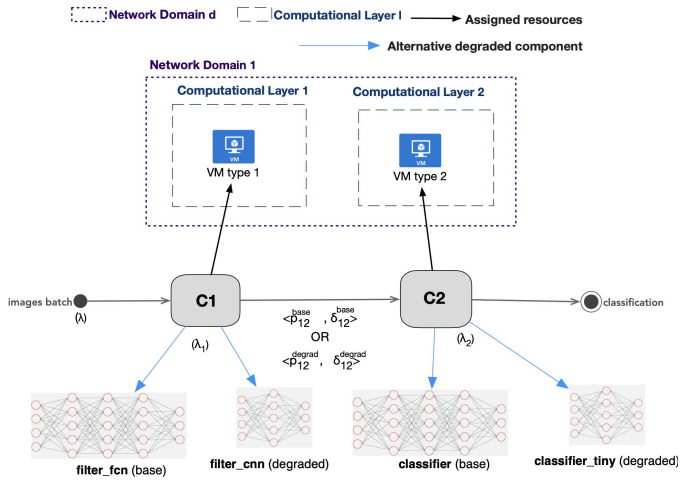
## Allows defining applications with degraded performance

- The user can provide multiple version of both filter and classifier components. Performance of the alternative applications is automatically computed and applications are ordered accordingly.
- SPACE4AI-R Runtime tool is able to switch from one alternative workflow to another in the case is needed

alternative_workflows.yaml

```yaml
System:
  name: filter_classifier_app
  alternative_dependencies:
    - alternative_1:
        dependency: [['filter', 'classifier', 1]]
        metric:
          name: average_f1
          value: 0.9
    - alternative_2:
        dependency: [['filter_v2', 'classifier_v2', 1]]
        metric:
          name: average_f1
          value: 0.85

    - alternative_6:
        dependency: [['filter_v3', 'classifier_v3', 1]]
        metric:
          name: average_f1
          value: 0.6
```



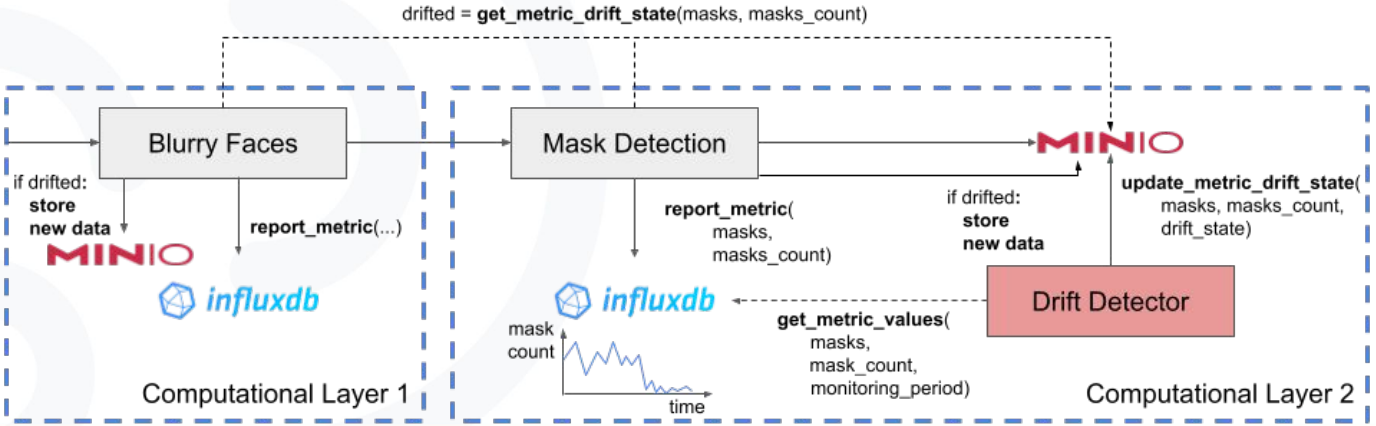
# AI-SPRINT Quality of Service (QoS) Annotations

| Annotation name |
|---|
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| *@aisprint.early_exits_model* |
| *@aisprint.model_performance* |
| ***@aisprint.detect_metric_drift*** |
| *@aisprint.security* |

```
def detect_metric_drift(metric, field, statistical_test,
                        test_threshold,detection_interval,
                        monitoring_period, data_collection_period)
```

Allows detecting data drift at runtime, by triggering the automatic deployment of the **Drift Detector** tool

- Periodically queries user-defined metrics (stored in InfluxDB) and run statistical algorithms to detect changes in the time series
- Allows collecting new data after the drift to be used for re-training the DNN-based components



# AI-SPRINT Quality of Service (QoS) Annotations

| Annotation name |
| --- |
| *@aisprint.component_name* |
| *@aisprint.exec_time* |
| *@aisprint.expected_throughput* |
| *@aisprint.partitionable_model* |
| *@aisprint.device_constraints* |
| *@aisprint.early_exits_model* |
| *@aisprint.model_performance* |
| *@aisprint.detect_metric_drift* |
| ***@aisprint.security*** |

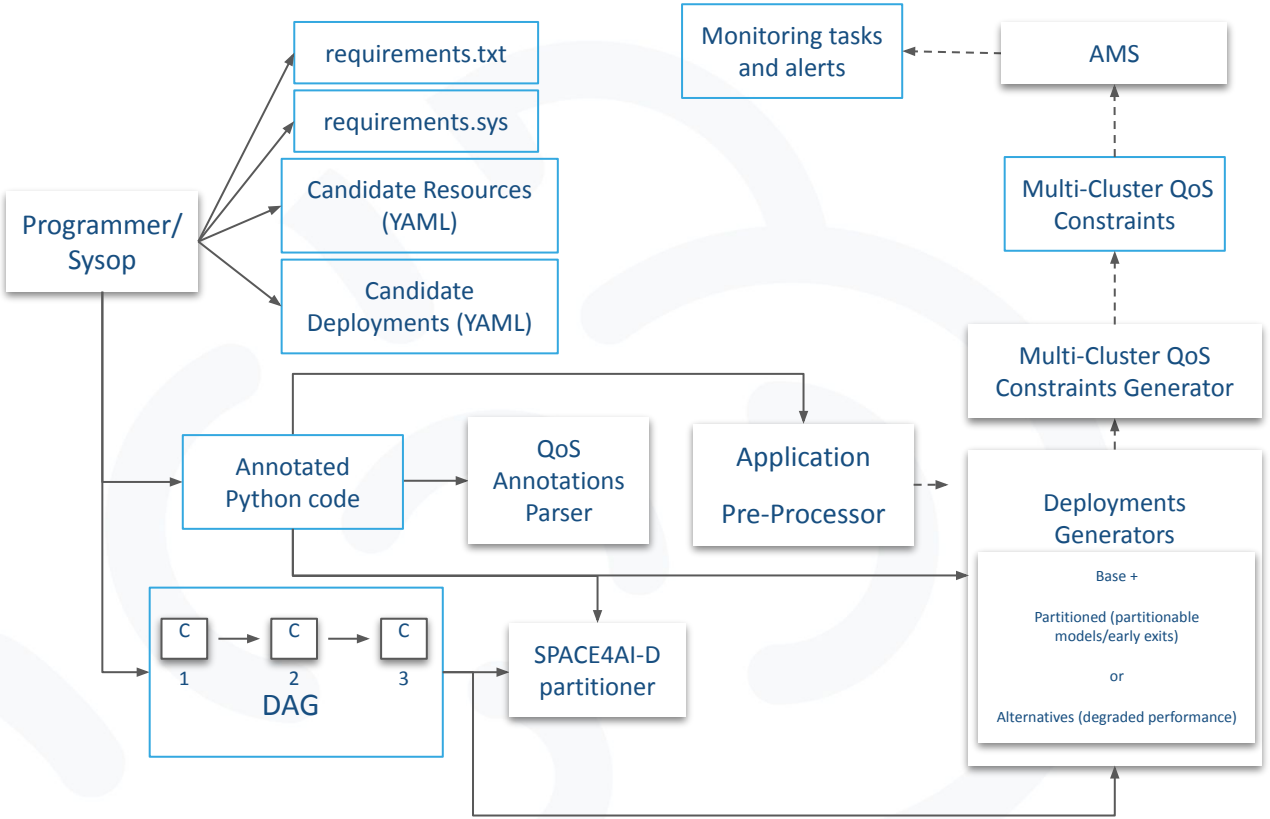
```
def security(trustedExecution, networkShield, filesystemShield,
             confProc, integrityProc, confRest)
```

Allows users annotating components to receive security guarantees while executing:

- Enables *trusted execution environments* and secure boot
- Wraps TCP connections using the **SCONE** network shielding layer
- Enables the encryption of all the files written and read by the process executing the task

# AI Application Design Workflow

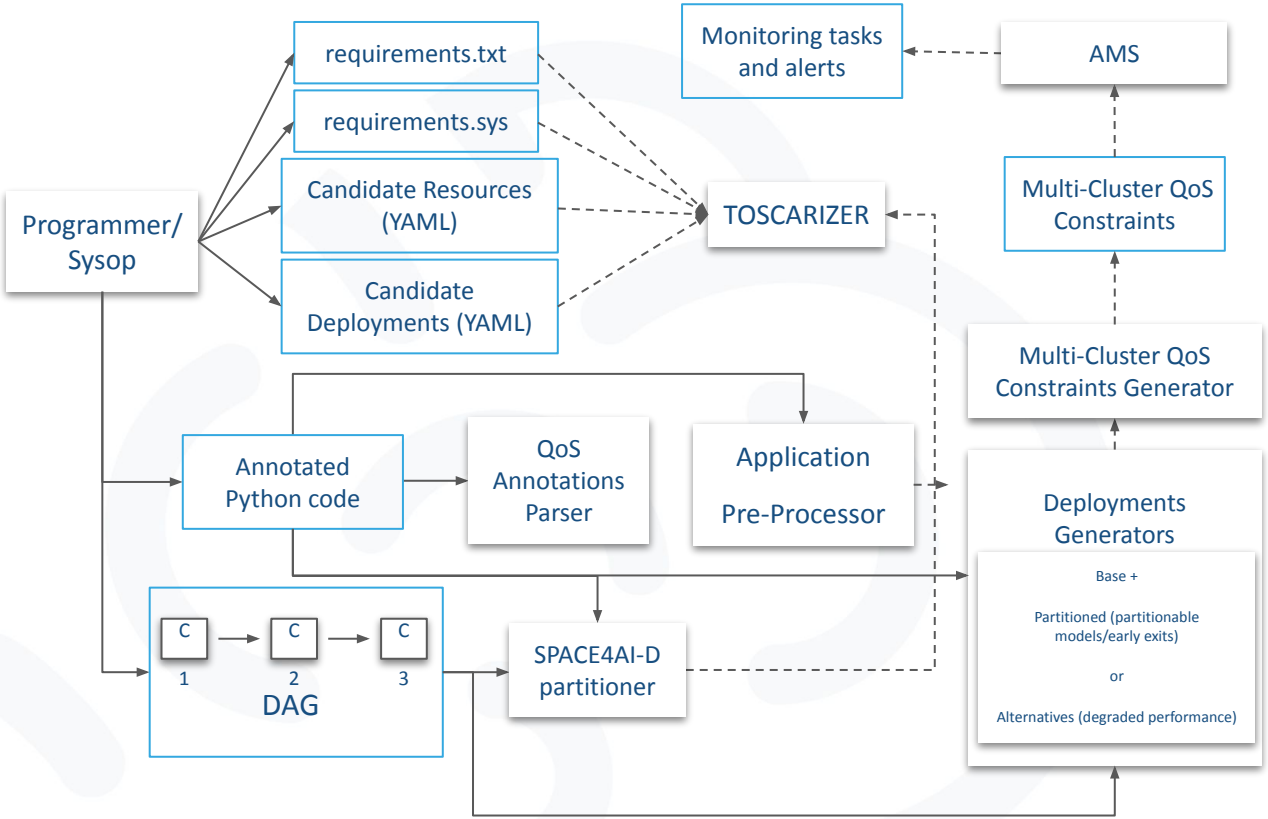
## Run AI-SPRINT Design with AI-SPRINT Studio https://gitlab.polimi.it/ai-sprint/ai-sprint-studio




```yaml
deployment2:
  ExecutionLayers:
    1:
      components:
      - blurry-faces-onnx
      local_constraints: {}
      global_constraints: {}
      throughput_component: blurry-faces-onnx
    2:
      components:
      - mask-detector-onnx_partition1_1
      local_constraints: {}
      global_constraints: {}
      throughput_component: blurry-faces-onnx
    3:
      components:
      - mask-detector-onnx_partition1_2
      local_constraints: {}
      global_constraints:
        global_constraint_1:
          path_components:
          - blurry-faces-onnx
          - mask-detector-onnx_partition1_1
          - mask-detector-onnx_partition1_2
          threshold: 30
      throughput_component: blurry-faces-onnx
deployment3:
  ExecutionLayers:
    1:
      components:
      - blurry-faces-onnx_partition1_1
      local_constraints: {}
      global_constraints: {}
      throughput_component: blurry-faces-onnx_partition1_1
    2:
      components:
      - blurry-faces-onnx_partition1_2
      - mask-detector-onnx
      local_constraints: {}
      global_constraints:
        global_constraint_1:
          path_components:
          - blurry-faces-onnx_partition1_1
          - blurry-faces-onnx_partition1_2
          - mask-detector-onnx
          threshold: 30
      throughput_component: blurry-faces-onnx_partition1_1
```

# AI Application Design Workflow

## Run TOSCARIZER with AI-SPRINT Studio https://gitlab.polimi.it/ai-sprint/ai-sprint-studio



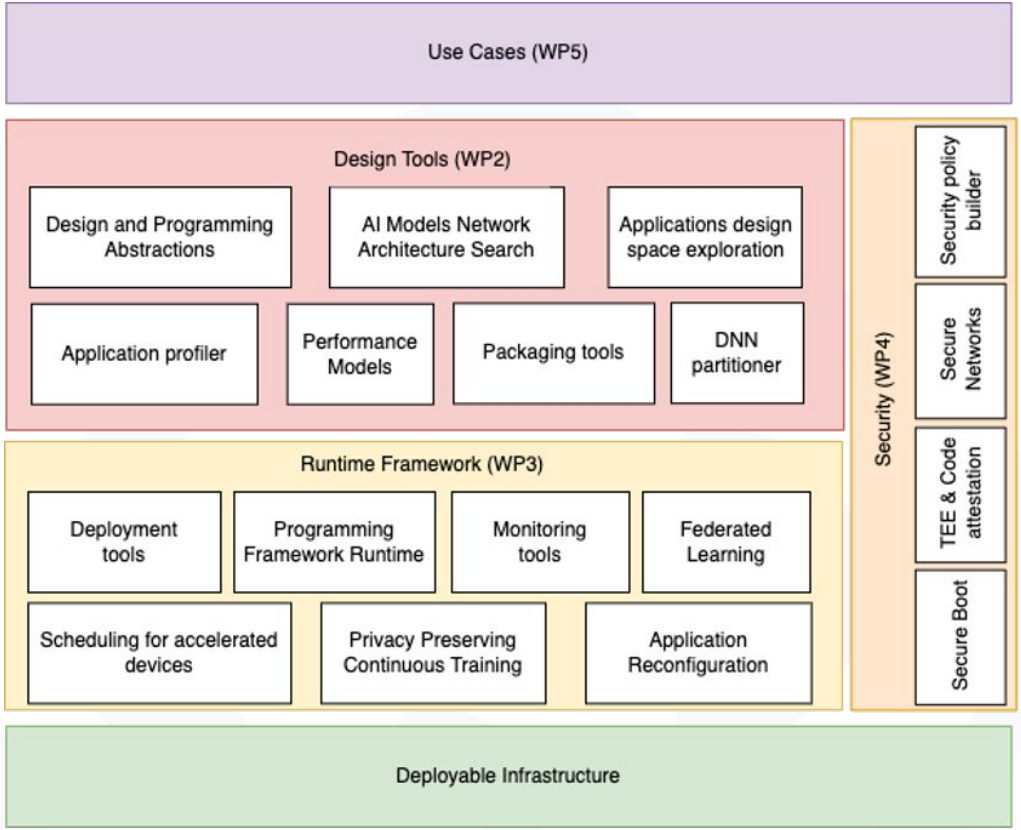

**Tool** to help Application Manager to deploy Inference services.

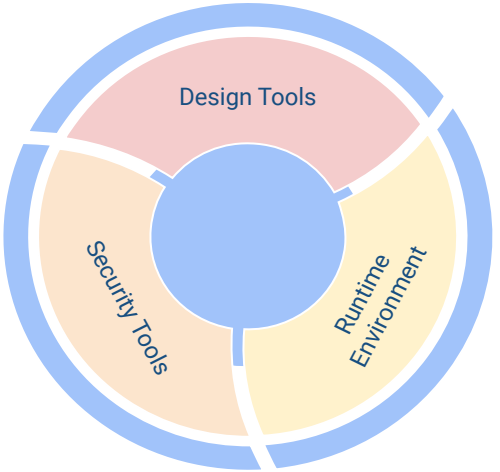
- **Creates the Docker images** for all application components considering all possible destination architectures (AMD64 and ARM64).

Furthermore

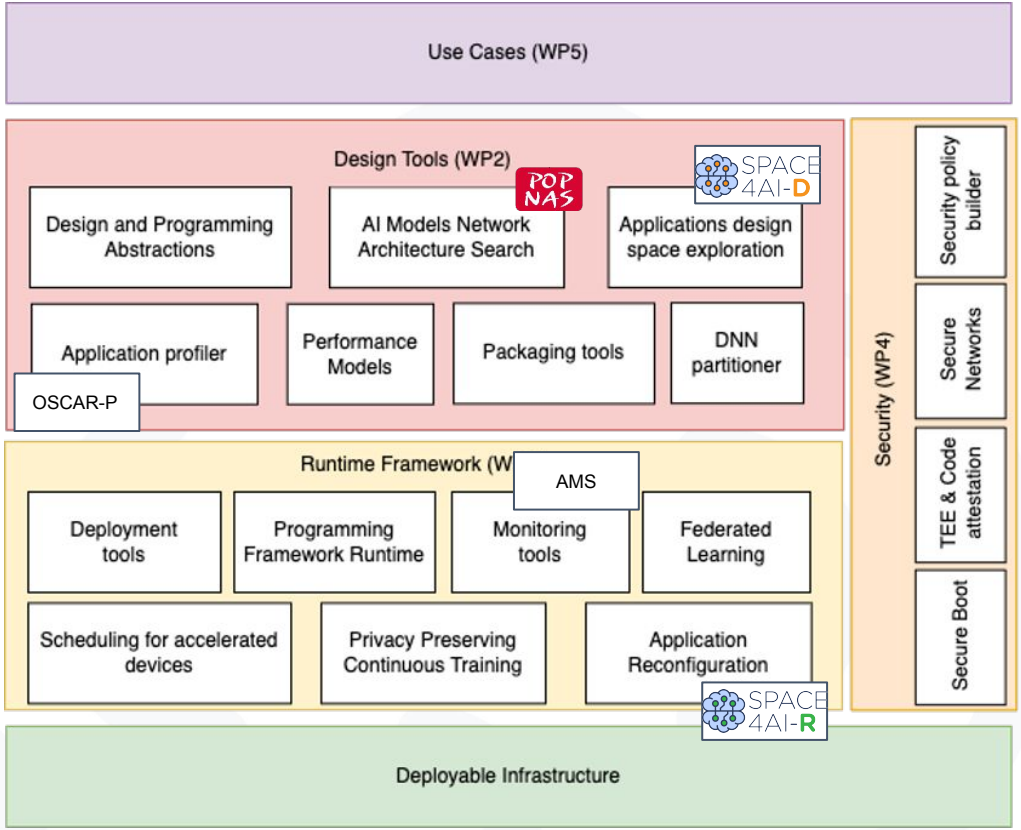
- **Creates TOSCA templates** to deploy, not only the inference services on top of OSCAR clusters, but also all the needed underlying cloud infrastructure (VMs, K8s cluster, OSCAR …).
  - Deploys the full application workflow.
- **Interacts with the Infrastructure Manager (IM** )to finally deploy/undeploy all the inference infrastructure and get the endpoints of the deployed services.

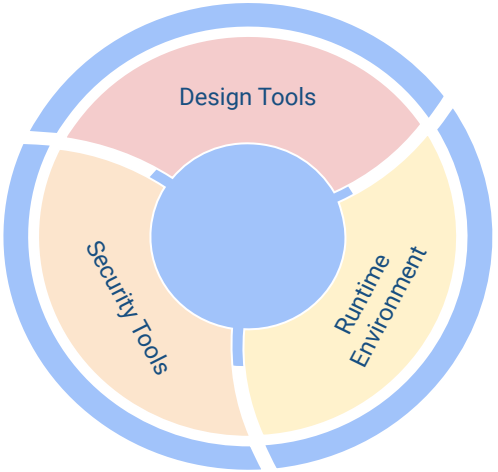
# The Big Picture



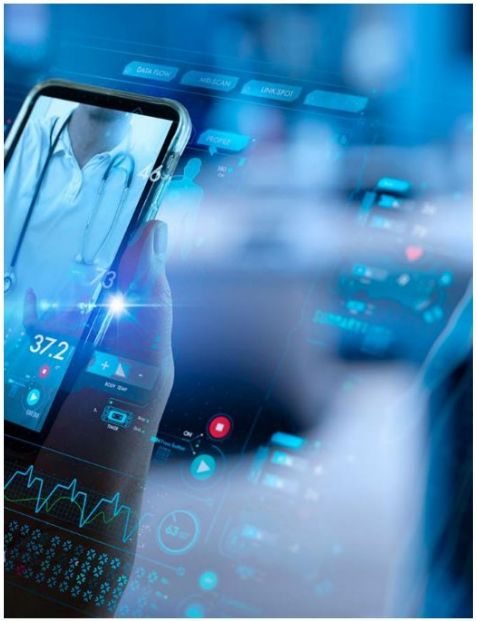



# The Big Picture



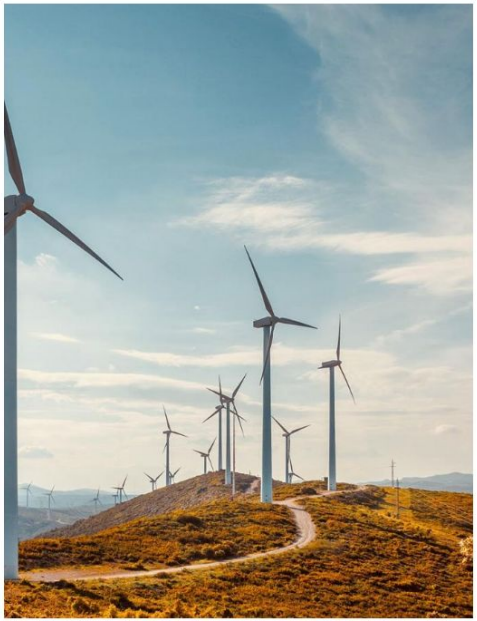



# AI-SPRINT Use Cases Overview



## Personalised Healthcare

Developing an automated system for personalised stroke risk assessment and prevention.



## Maintenance & Inspection

Creating an infrastructure that reduces downtime and revenue losses caused by degenerative asset performance.

## Farming 4.0

Delivering edge and intelligent sensors to optimise phytosanitary treatments.

# AI-SPRINT Alliance

The AI-SPRINT Alliance is composed of a group of a specialised supply and demand community of Software houses, AI-application developers, System integrators, Cloud Providers, Digital Innovation Hubs, and R&D initiatives that can use the **AI-SPRINT components and tools with technical support** from the project's partners

# AI-SPRINT Consortium

COORDINATOR

POLITECNICO MILANO 1863

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

GRE OIRE

Beck et al. work. together.

CLOUD & HEAT

7bulls.com

TTAnalysis

Trust-IT Services
communicating to markets

IDC

Thanks for your attention!

AI-SPRINT demos available at: https://gitlab.polimi.it/ai-sprint/ai-sprint-examples
AI-SPRINT Studio library available at: https://gitlab.polimi.it/ai-sprint/ai-sprint-studio
AI-SPRINT Studio Docker image available at: registry.gitlab.polimi.it/ai-sprint/ai-sprint-studio