Distributed interactive systems using Ingescape

Part 1

UPSSITECH

September 2024

Distributed software systems

What is the current situation for software in operational systems ?

- Modern operational systems are **interactive**, **distributed** and **heterogeneous.**
- System integration is always a hassle... and should not be!
- Building digital environments **requires collaboration** between many stakeholders with different views, skills and objectives
  - system engineers, software developers, platform maintainers, end-users, designers, human factors specialists, domain experts, etc.
- A digital environment's **foundations** should be **strong** and its **structure** should be **flexible**, from early works to operational exploitation, with **continuous evolutions**.

The industry requires enterprise-grade interoperability for all software, legacy and new, involving all the relevant actors in software projects.

Why have distributed systems become so important ?

- **Geography**
  - Systems and data are now decentralized and distributed
  - Resources and software now spread on multiple locations
- **Heterogeneity**
  - Systems gather many technologies, making interoperability a major stake
  - Teams want to work with their own techniques & tools in agile dynamic contexts
- **Reusability**
  - Despite the ambient complexity, problems often repeat themselves, and so should the solutions
  - Service-oriented architectures are a real solution but often poorly designed or implemented
- **Collaboration**
  - Humans and machines are parts of the same interactive environments
  - Interactions always happen in parallel and on-the-fly, sometimes concurrently

State of the art for distributed systems

Client/server

Web services : HTTP, SOAP/REST, XML/JSON

SOA : Service Oriented Architectures

ESB : Enterprise Service Bus

Cloud : virtualization, containerization

Messaging & protocols

Apache kafka · MQTT · RabbitMQ

AMQP Advanced Message Queuing Protocol · Stomp · JMS

Apache ActiveMQ · MSMQ · Red Hat Fuse

MBSE / MDA

MBSE : Model-Based Software Engineering

MDA : Model-Driven Architecture

OMG Object Management Group · UML Unified Modeling Language · CORBA · DDS · XML WSDL · emf Eclipse Modeling Framework · SysML · ARCADIA

ZeroMQ, the communication layer under Ingescape

"ZeroMQ looks like an embeddable networking library but acts like a concurrency framework for distributed software. It gives you sockets that carry atomic messages. You can connect sockets N-to-N with patterns like fan-out, pub-sub, task distribution, and request-reply."

- Zero broker, zero latency, zero cost, zero administration

  « More generally, 'zero' refers to the culture of minimalism that permeates the project. We add power by removing complexity rather than by exposing new functionality. »

- Multi-languages, multi-OS
- Multi-transports (IPC, TCP, multicast, WSS, etc.)
- Active open source community

Ingescape...

Ingescape is already used in many industries

RATP

NATO OTAN

AIRBUS AN EADS COMPANY

sopra steria

Atos

EUROCONTROL

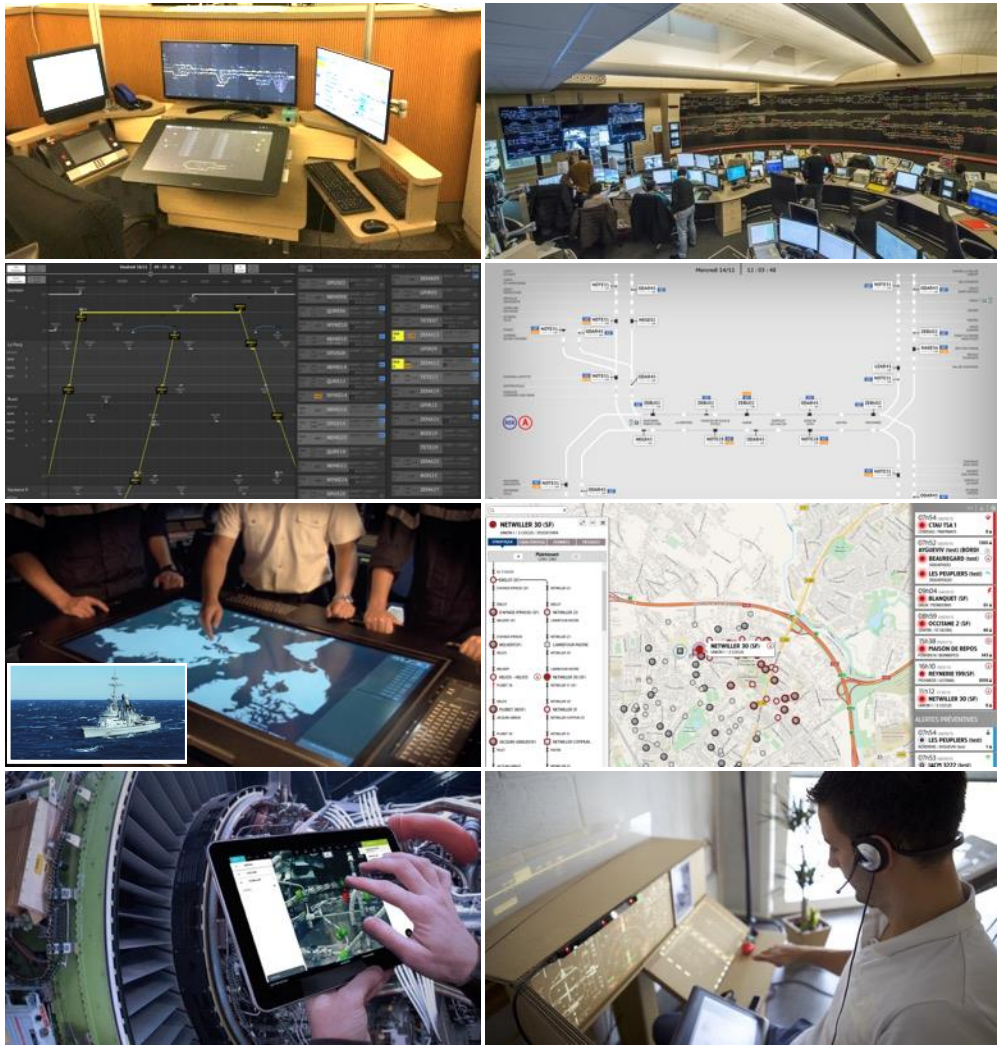
AIRBUS DEFENCE & SPACE

ENEDIS L'ELECTRICITE EN RESEAU

DGA DIRECTION GÉNÉRALE DE L'ARMEMENT

DASSAULT AVIATION

THALES



https://www.youtube.com/watch?v=9Gr1Le_F7jU

Full-scale software interoperability

INGESCAPE

Library

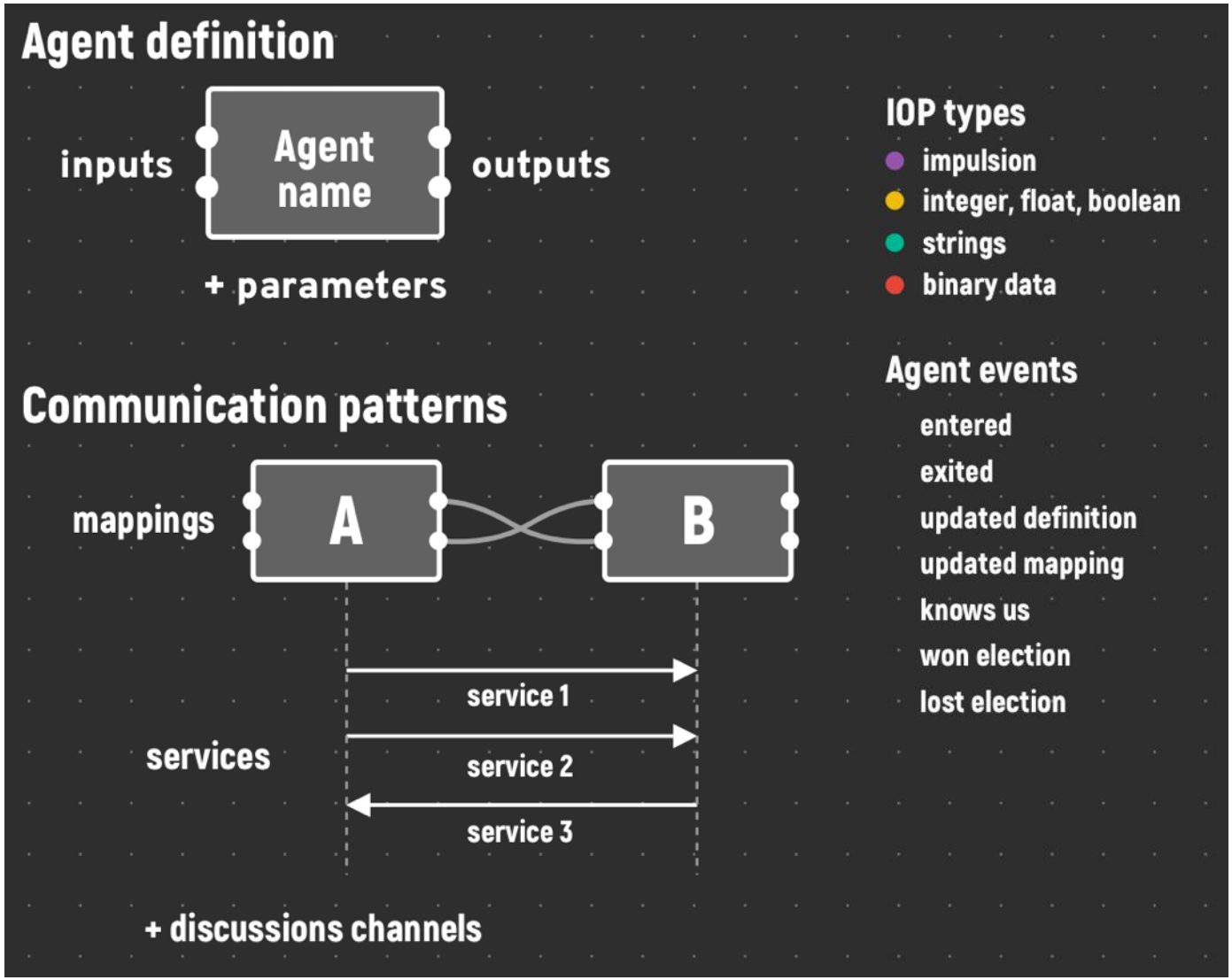
Any language, any OS, web, cloud, **open source**

Highly-supervised + fully-decentralized

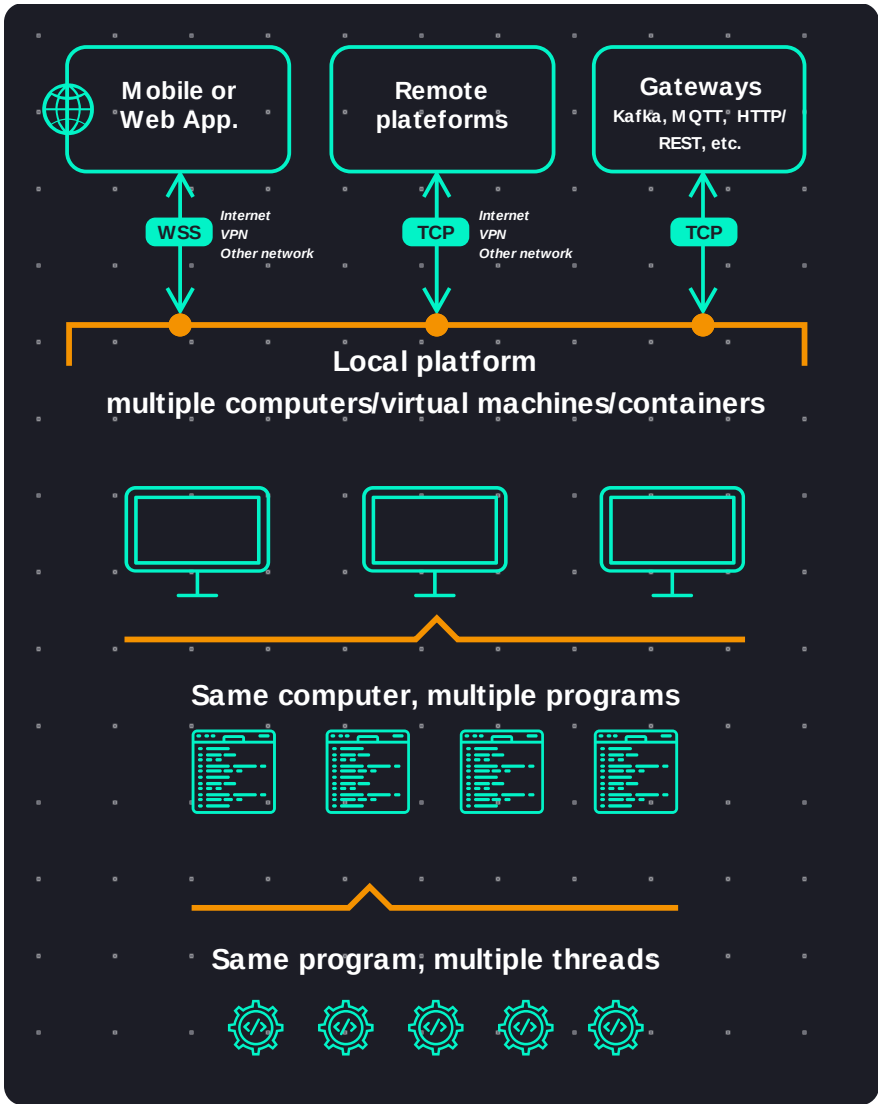
Model-based

https://github.com/zeromq/ingescape

The Ingescape concepts in a single picture



Scalability from worldwide systems to CPU-level high performance computing



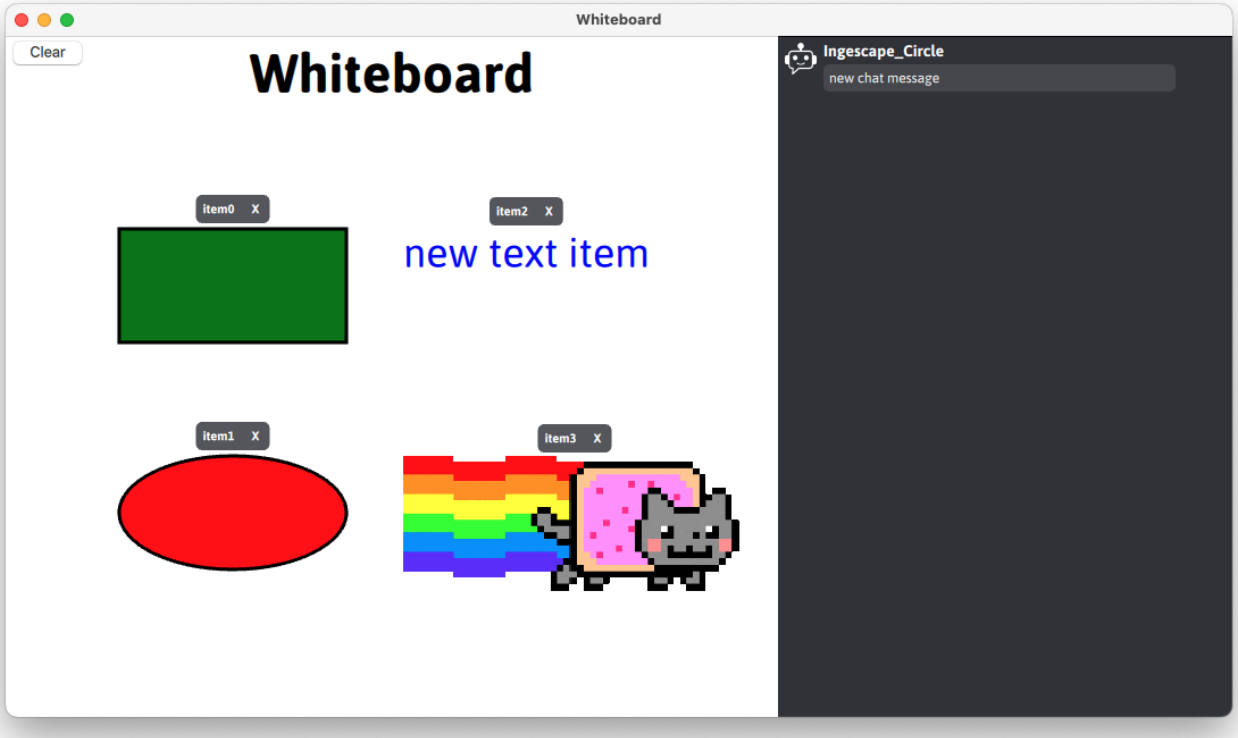
Your project

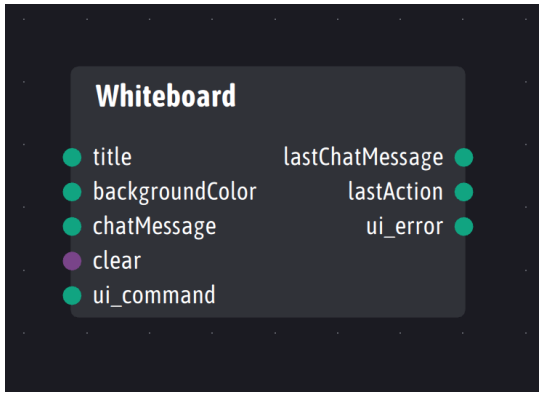
Students project philosophy

- Exploring interactive principles by the creation of a system of systems involving the contribution of 15 groups of 2 students each
- Enabling groups to collaborate between them and with the teachers
- Applying good industrial practices
  - Iterative prototyping and development
  - Specification and model-based V&V
- Reaching an actual result in record time (around 1 month)
- Ultimately making you touch what it takes to design an interactive system that actually works

We provide one agent…

- The WhiteBoard application
  - Fully described in the ***WhiteboardDemo.igsplatform*** platform file





...you provide the others around it

A	**Android app** Send pictures, add text and draw figures to the white board. Show what is on the white board and enable real-time interaction.
B	**Speech recognition and synthesis** Convert vocal commands to interact with the white board. Tell what happens on the white board.
C	**AI-based image generation** Text-based image generation sent as image to the white board. Image generator as a service from the white board chat.
D	**VR or Web client** Display the content of the white board. Provide interactions and ability to contribute (text, geometry, images) in real-time.
E	**IoT environment** Use IoT devices and hardware to enrich the white board experience by notifying events and enabling physical interactions.
F	**Chatbot** Conversational agent to interact with the white board, both to query its state and event, and to contribute to the content (text and geometry).

- Or you can propose your own agent...

Evaluation criteria

- Quality of the proposed User eXperience **/5**
  - Utility, efficiency, comfort, robustness
- Completeness of the integration with the white board **/5**
  - Use of the white board's inputs, outputs and services in your own agent
  - Bonus points if you interact with other agents for an extended user experience.
- System engineering **/5**
  - Agent requirements
  - Minimal specifications for your agent (less is more)
  - Complete V&V scripts with traceability to your requirements
- Coding **/5**
  - Documentation
  - Ability for the teachers to compile and run the code
  - Clarity, concision and robustness

Calendar

Friday September 27th	• Groups formation (2 students per group) • Each group registers by sending an email to upssitech@ingenuity.io with the students names and subject chosen.
Friday October 18th	• 1st practical work session, assisted by the Ingenuity team • Technical choices, compilation, debug environment • First igsplatform for basic test and debugging
Monday November 4th	• 2nd practical work session, assisted by the Ingenuity team • Continuous testing, V&V scripting, live integration
Tuesday November 5th	• Last practical work session, assisted by the Ingenuity team • Integration and testing with the whiteboard and other agents
Monday November 25th	• Project delivery to upssitech@ingenuity.io (less than 9MB) or Github • Documentation, ingescape platform for integration and tests, V&V scripts, source code, compiled code

The (multidisciplinary) team involved in this course

Stéphane

Alex

Mathieu

Chloé

Aurélien

Where to get Ingescape and other resources ?

- The open source Ingescape **library repository**
  - https://github.com/zeromq/ingescape
- The Ingescape **Circle installer**
  - https://ingescape.com/get
- The **license and resources for this course**
  - https://ingescape.com/upssitech
- The open repository for the **Whiteboard agent**
  - https://gitlab.ingescape.com/learn/whiteboard