Human Computer Interaction course

Part 1

ENSEEIHT

January 2023

WARNING

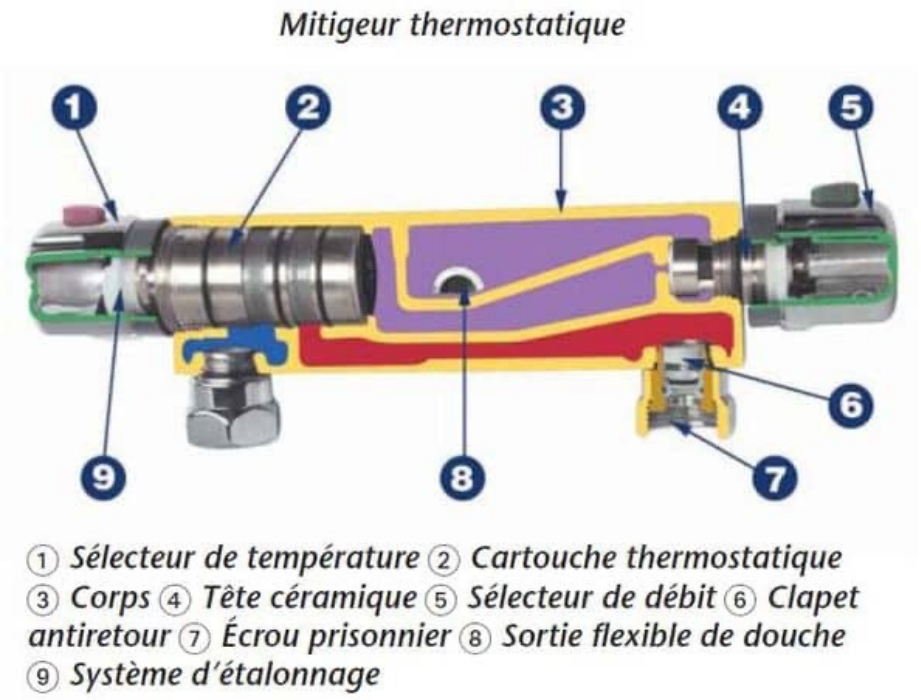
- This course heavily relies on interactive demos, which are not in the slides. Your attendance is recommended.
- Your exam will consist in developing a small interactive application with technologies you will discover in this course.
- Tutorial sessions will be dedicated to helping you develop your application.

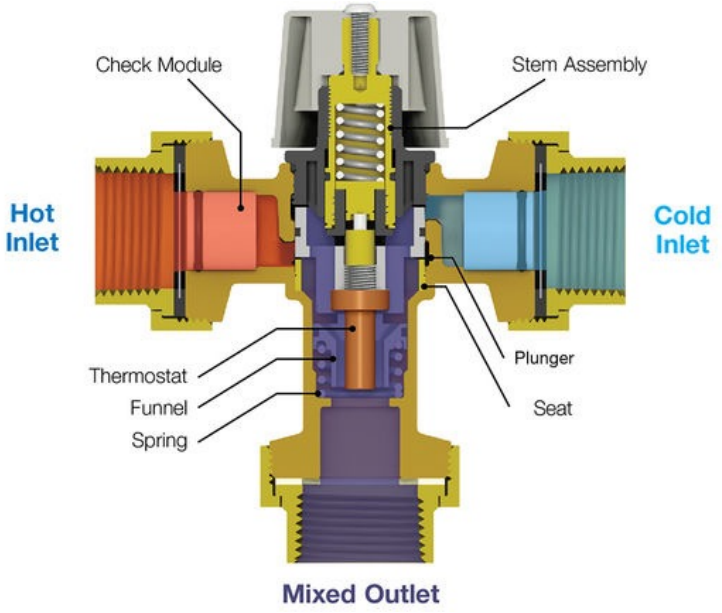
Course content

- Course 1
  - HCI, HSI, distributed systems, interactive software engineering
  - First contact with Ingescape
  - Presentation of the exam
- Course 2
  - Exam groups
  - HCI & UX methodologies
  - Visual programming with Ingescape
- Course 3
  - Software architecture for HCI development
  - Generating code and crossing models for interactive applications
  - Verification & Validation applied to interactive systems
- Course 4
  - Methodologies for multidisciplinary and iterative System Engineering, notions of HSI
  - Human Factor assessments, why and how
  - Co-simulation and data record/replay with Ingescape
- Course 5
  - Practical exchanges on your exam projects using system architecture models

The essence of HCI design

- Do you take showers ?





- Identify relevant system data, data to be displayed, and data to be manipulated by the users
- Define the logic between system data and user data (a.k.a. HCI functions)
- Find the best forms for data presentation and manipulation

Which expertise and methodology for HCI ?

System
+
Human

Multidisciplinary

- System expertise = engineers
- Human expertise = ergonomists
- Creators = designers & engineers

Analysis
+
Creativity
+
Evaluation

Iterative

- Express the problems
- Create and improve solutions
- Evaluate the problems & solutions

User-centered

- Human is the key to acceptability and performance
- A system is only a tool, even with A.I. and automation

System users are not only the “end-users”

- Any system has to be
  Designed
  Built
  Validated
  Used
  Maintained
  Disposed

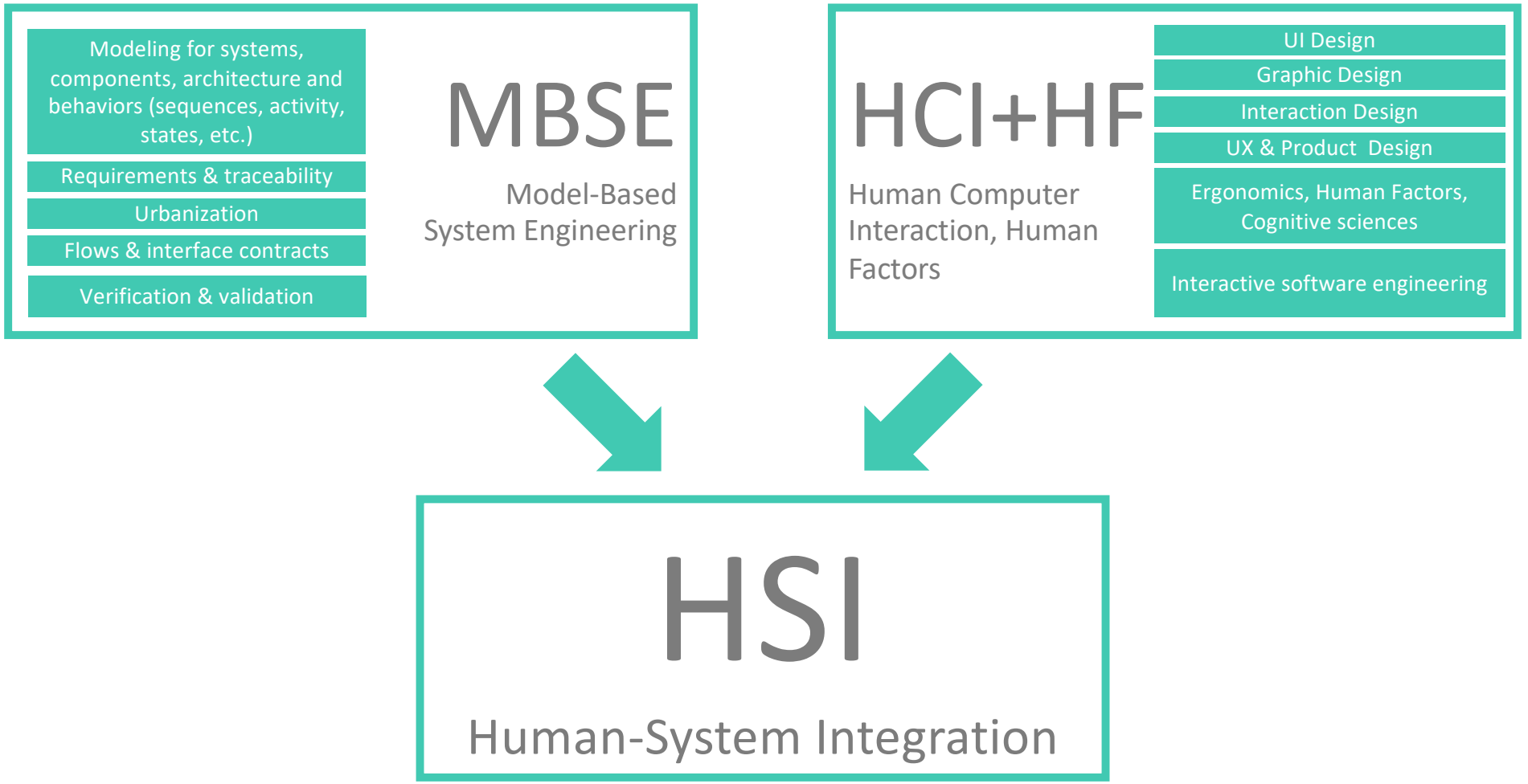
All these steps require users in the loop, who interact with the system in various ways.

The key is to maintain consistency at all stages of the system lifecycle.

Shocking revelation: This is not done very well in most industries and most companies, because people lack

- A shared & evolutive vision, supported by a “common language”
- Really collaborative environments, practices and tools
- Properly trained managers and organizations

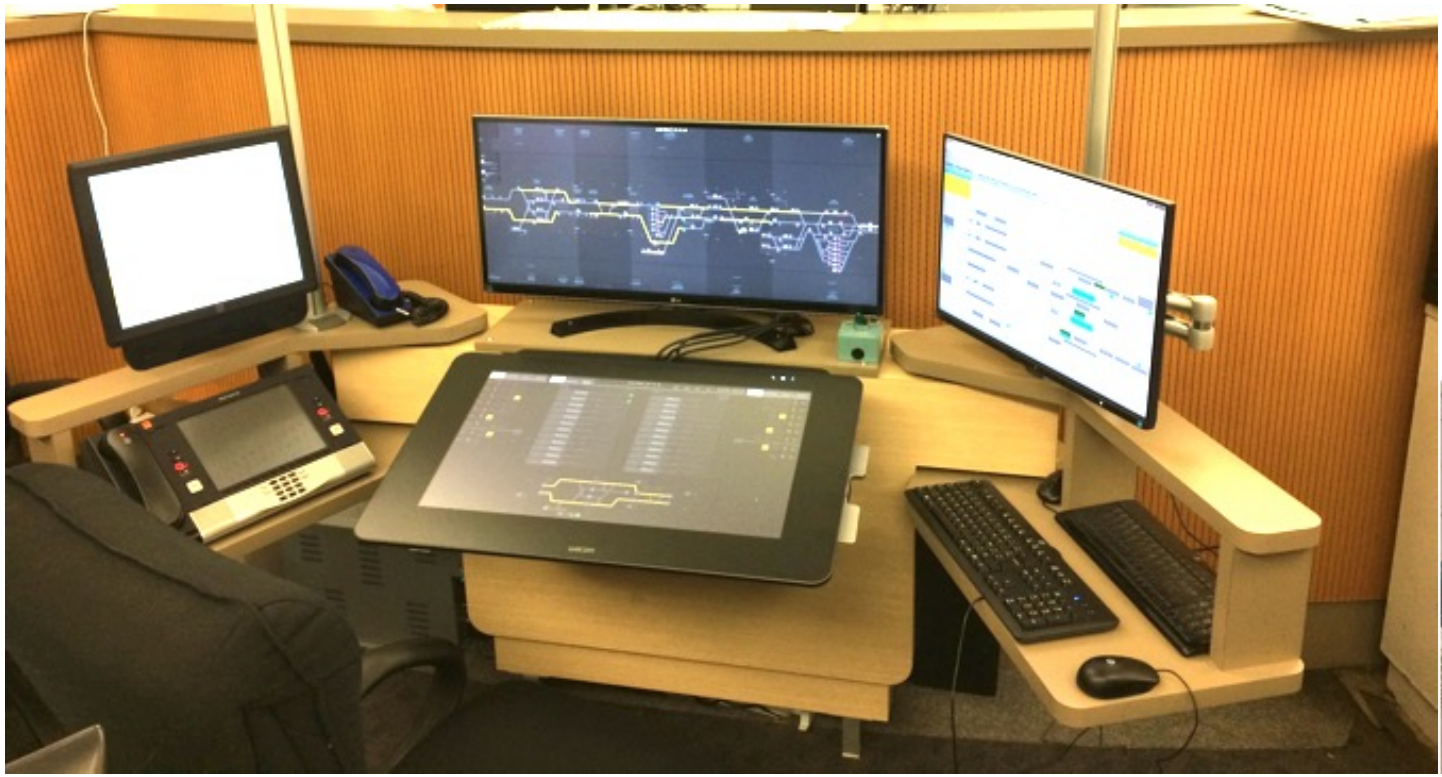
A comprehensive approach to engineering interactive systems



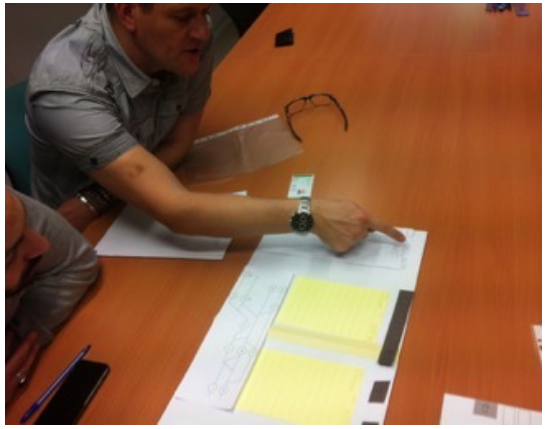
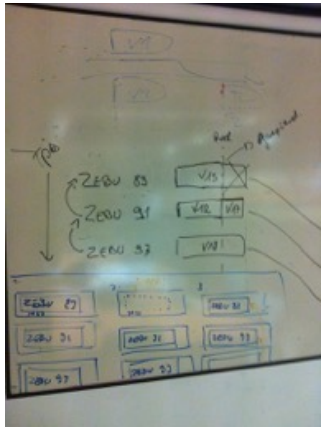
Design, development and deployment of interactive operational environments

Ingenuity io

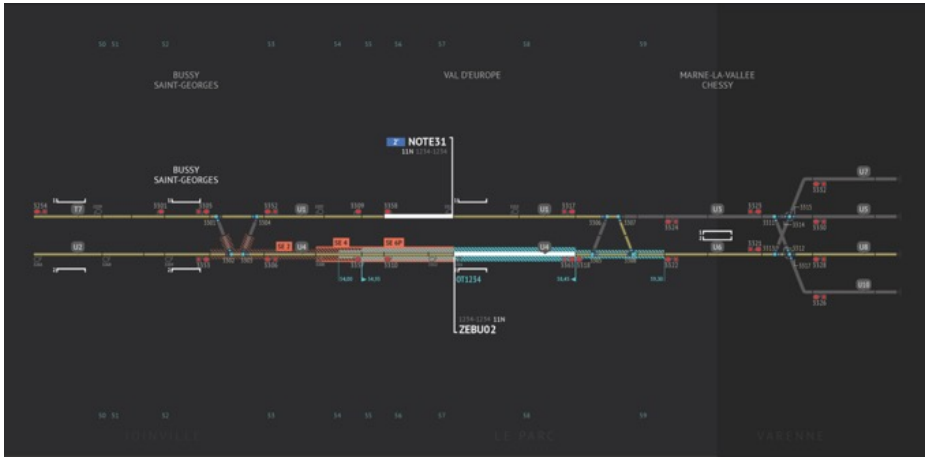
Full-scale real-time supervision and regulation for RER A & B

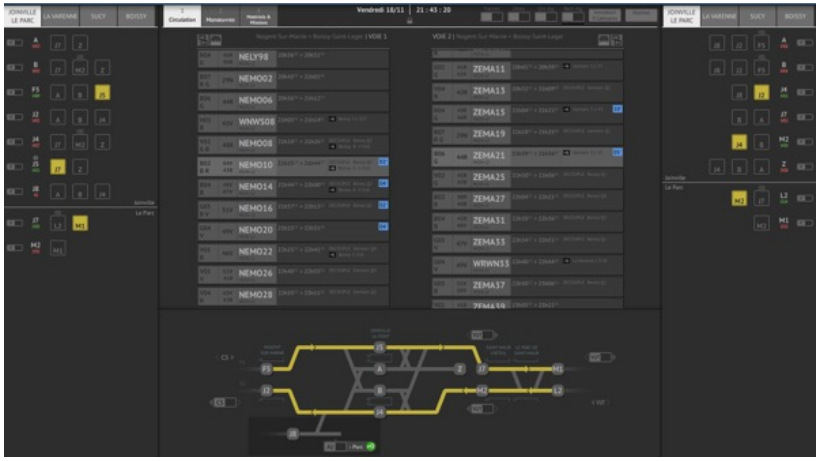


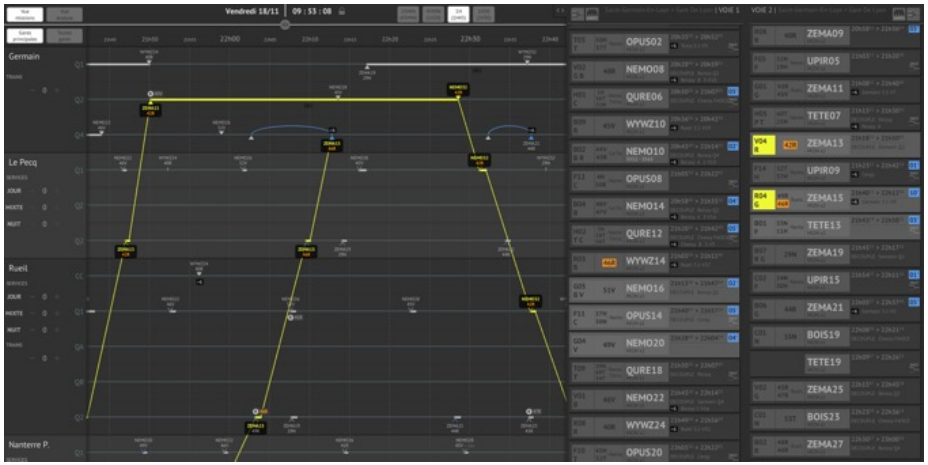
Full-scale real-time supervision and regulation for RER A & B

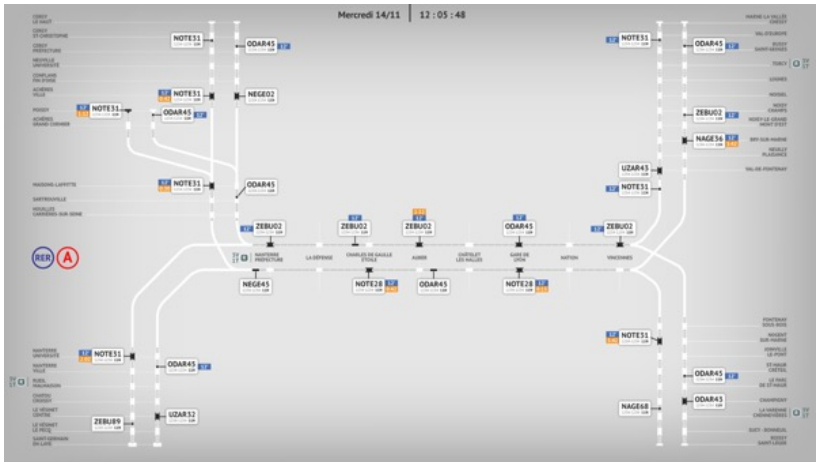





Full-scale real-time supervision and regulation for RER A & B



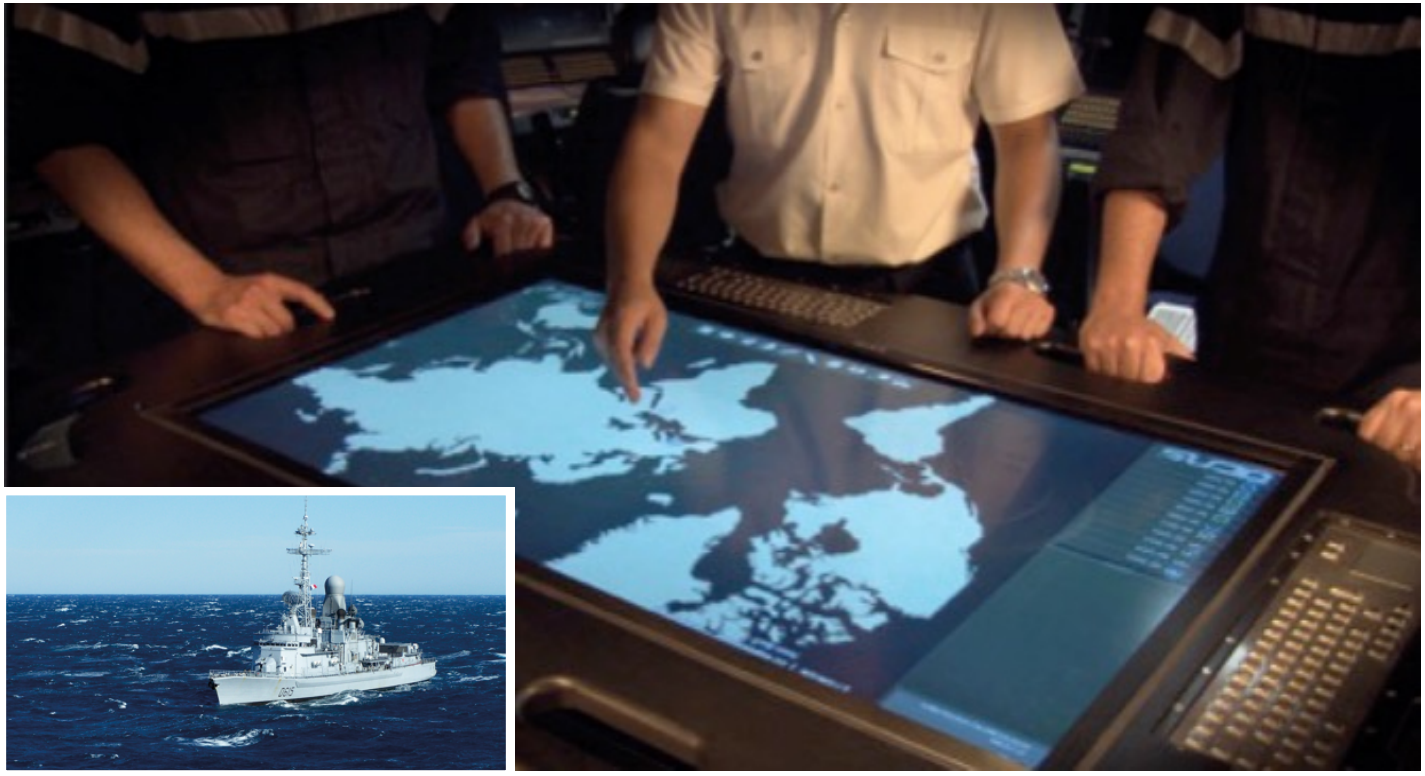






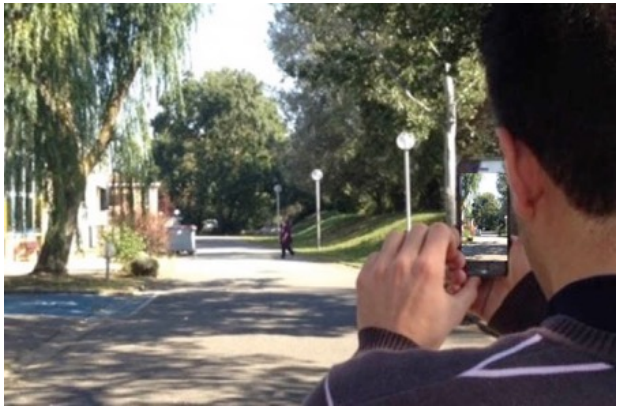
NAVAL GROUP

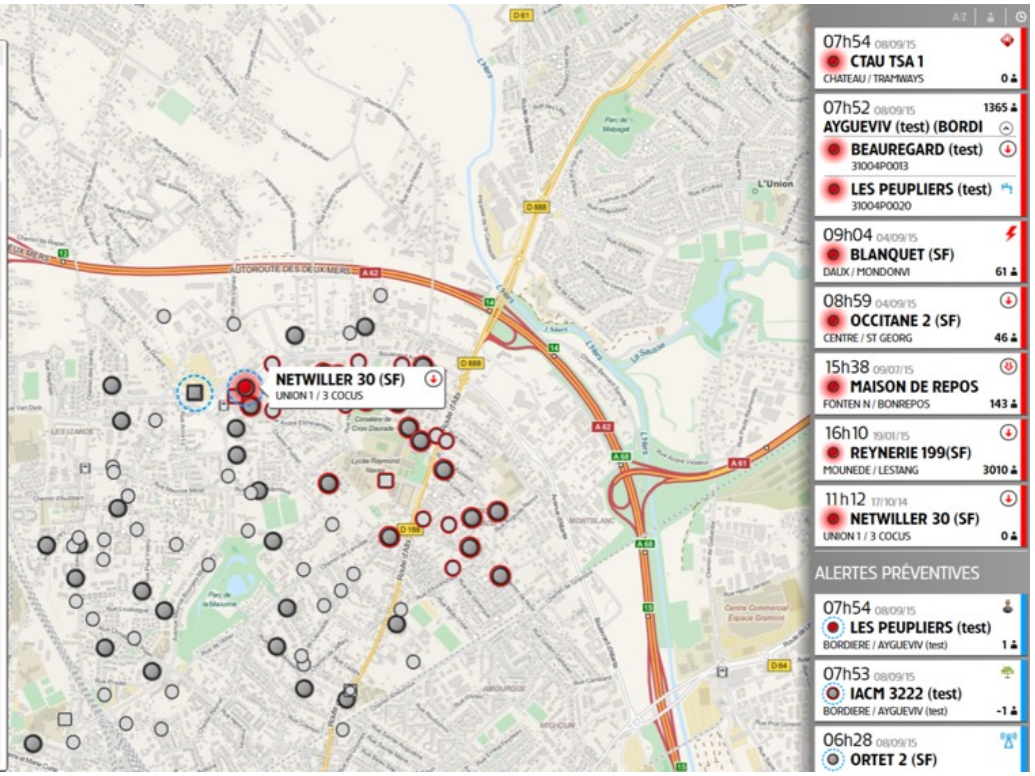
Collaborative table for tactical operations embedded in military ship's operational center



ENEDIS
L'ELECTRICITE EN RESEAU

Smart grid supervision with thousands of IoT devices, Big Data and augmented reality

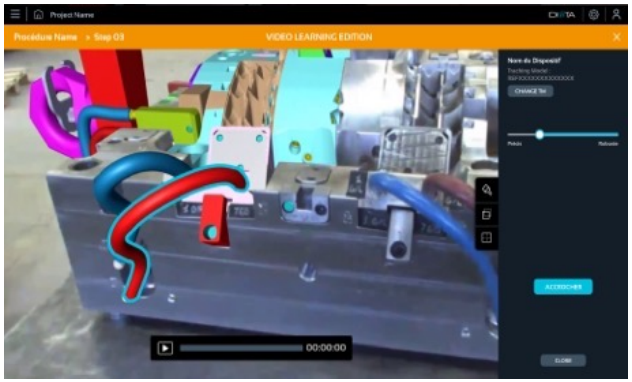


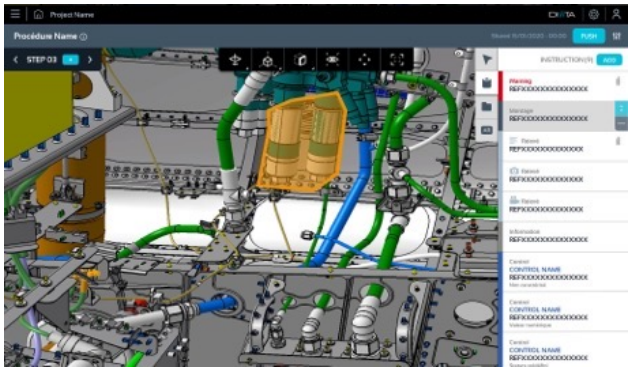


DiOTA

Digital-Field Solutions for Industry

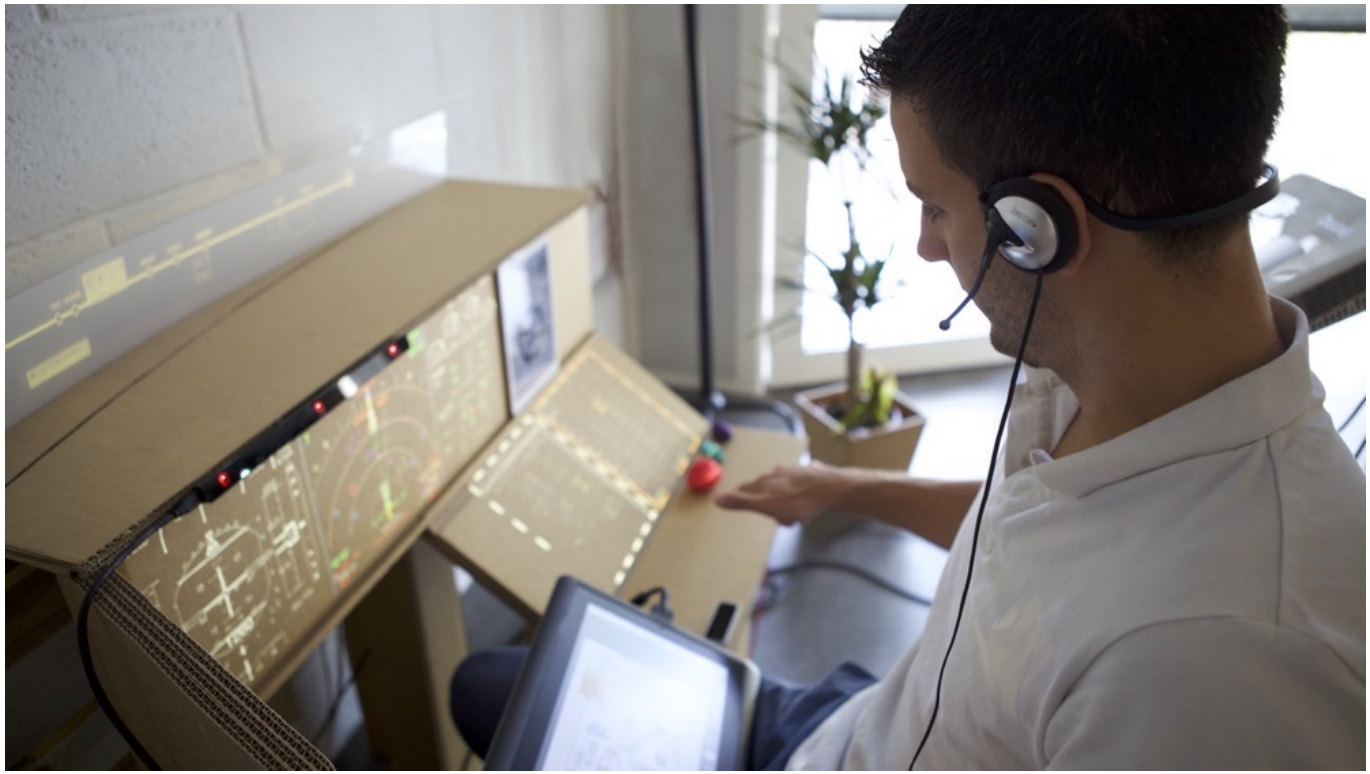
Live guidance and collaboration for industrial assembly and quality control, using 3D and augmented reality





AIRBUS

Advanced cockpit prototyping supporting state-of-the-art iterative and multidisciplinary methodologies



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

How did we handle these projects ?

- We have structured multidisciplinary collaborations in software teams and (re)conciliated agile, user-centered and iterative methodologies.
- We have developed the technologies and tools to support and accelerate the resulting processes.

INGESCAPE

Distributed software systems

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

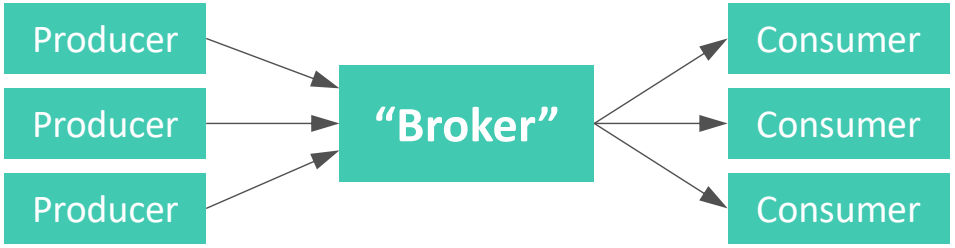
Web services : HTTP, SOAP/REST, XML/JSON

SOA : Service Oriented Architectures

ESB : Enterprise Service Bus

Cloud : virtualization, containerization

Messaging & protocols



MBSE / MDA

MBSE : Model-Based Software Engineering

MDA : Model-Driven Architecture

Ingescape...

Full-scale software interoperability

INGESCAPE

Library

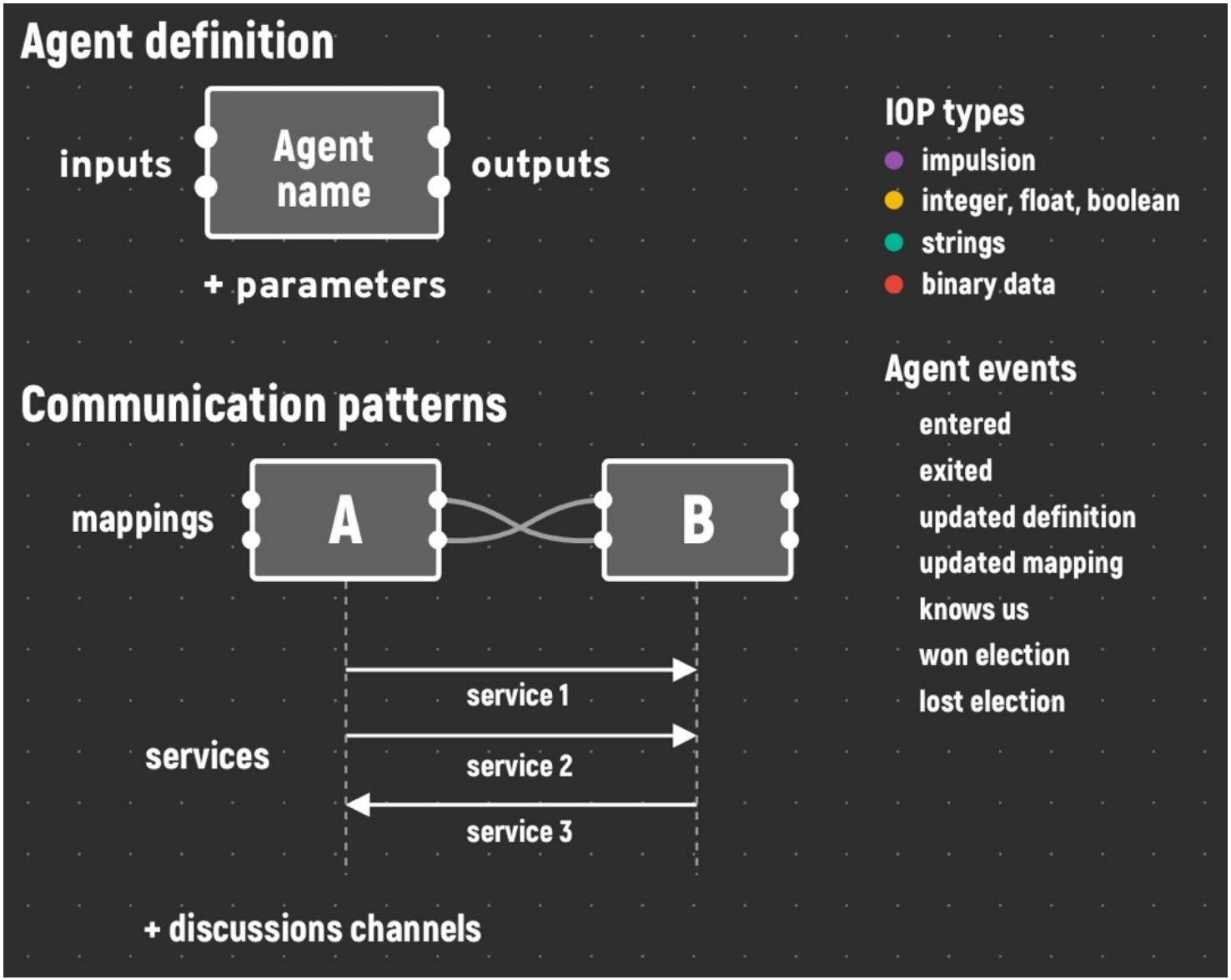
Any language, any OS, web, cloud, **open source**

Highly-supervised + fully-decentralized

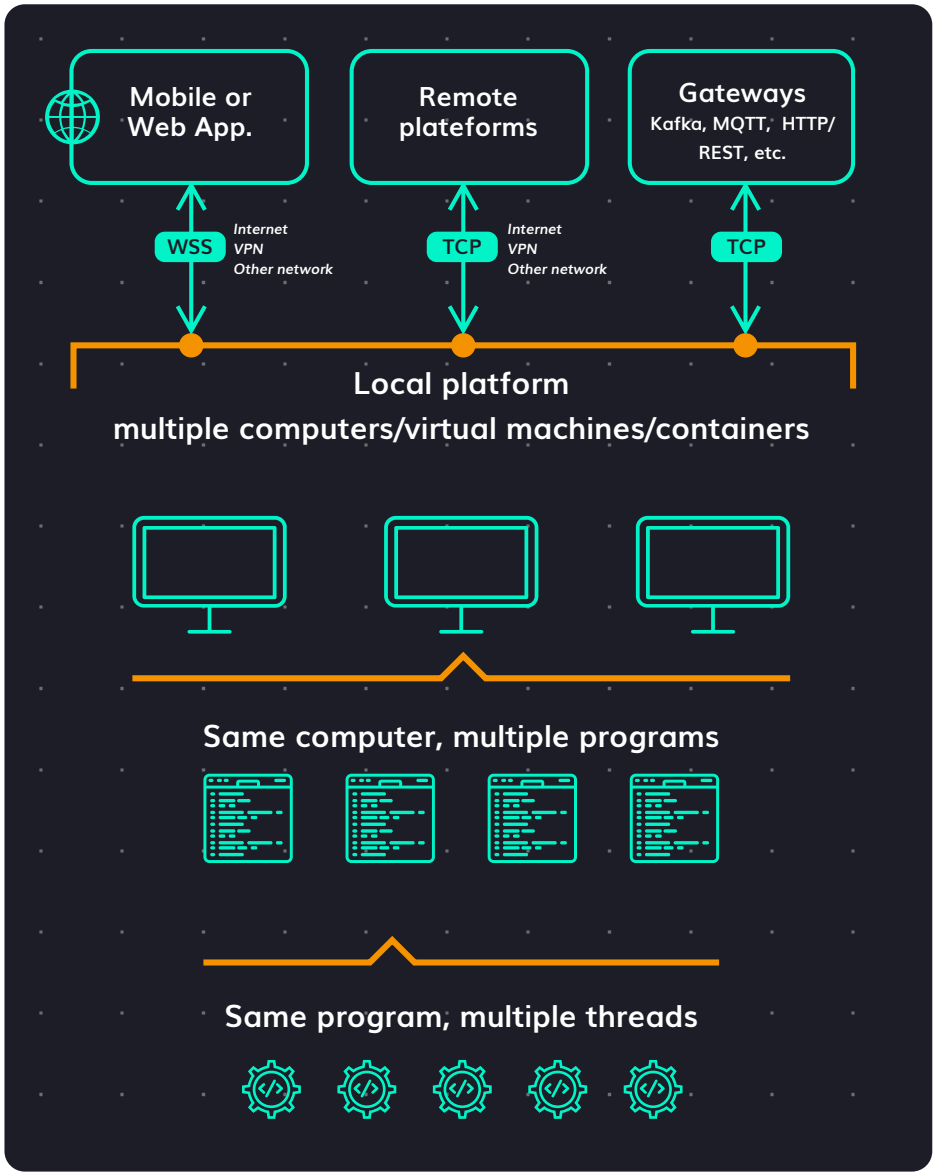
Model-based

https://github.com/zeromq/ingescape

The Ingescape concepts in a single picture



Scalability from worldwide systems to CPU-level high performance computing



A cornerstone for all phases in an industrial project

INGESCAPE Circle

- Mix models for architecture, software classes, network protocols, UI graphics, etc. to generate fully customizable code, preserving your preferred languages and frameworks
  - ⇒ Use code generation to bootstrap apps without losing control on your code
  - ⇒ Architecture model remains embedded in the code
- Give developers efficient visual tools that help them on a daily basis
  - ⇒ Enable iterative coding with active architect/developer collaboration
- Continuous model-based testing at unit, local and global levels
  - ⇒ Enable collaborative & constructive testing, avoiding surprises and tensions
  - ⇒ Open to iterative automatable Verification & Validation (V&V)

Ingescape is OPEN

- Open source
  - The Ingescape library is hosted on ZeroMQ's github pages as a full part of the ZeroMQ community, under the MPLv2 license
  - The Ingescape security layer is based on the libsodium encryption library
- All Ingescape models are transcribed in JSON
  - Easy to read, parse and generate
  - Open to third-party and home-made tools
- Ingescape embeds or instantly connects to many industry standards
  - Protobuf for serialized binary data
  - MQTT, Apache Kafka, RabbitMQ and other software buses
  - HTTP/REST, websockets, TCP, UDP and many other protocols
- Ingescape already provides bootstrap projects for
  - Cmake, Qt, Visual Studio, Xcode, NodeJS, etc.
  - C/C++/Obj-C, C#, Python, Java, HTML/CSS/JS, etc.
- Ingescape Circle is customizable and extensible
  - with any kind of specialized tools and modules that would address specific or generic needs.

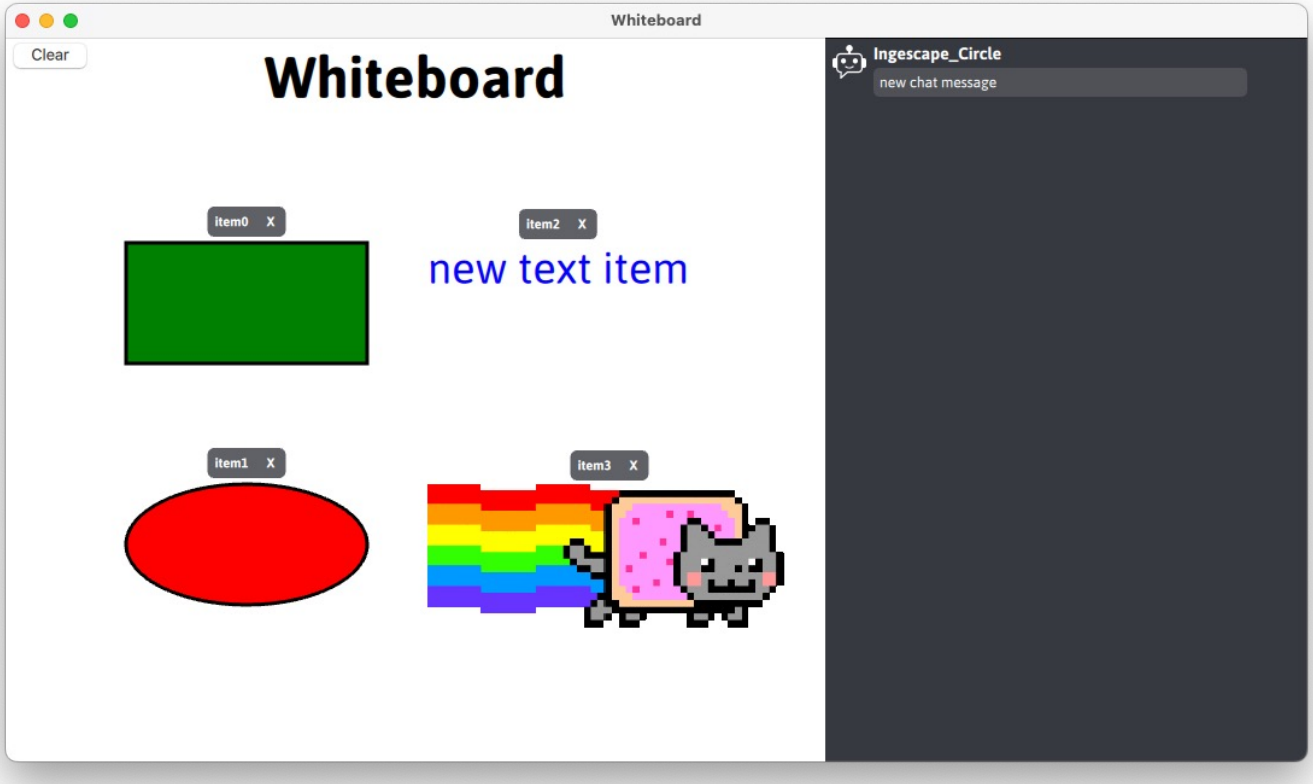
Your HCI project

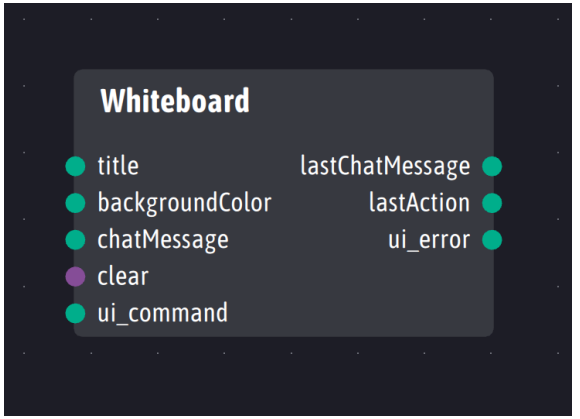
Students project philosophy

- Exploring HCI/HSI principles by the creation of a system of systems involving the contribution of 18 groups of 3 students each
- Enabling groups to collaborate between them and with the teachers
- Applying good industrial practices
  - Multidisciplinary design (UX, MBSE, software) with a user-centered approach
  - 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

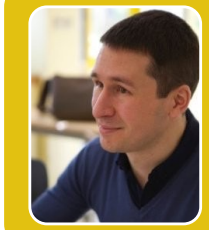
Calendar

Tuesday January 3rd	• Groups formation (3 students per group) • Each group ranks the 6 projects by order of preference filling and sending the excel sheet at https://ingescape.com/n7 • Groups and rankings sent to n7@ingenuity.io
Wednesday January 6th	• Groups announcements
Monday January 16th	• Interactive course with presentation of the system architecture and agent definitions
Friday January 20th	• 1st practical work session, assisted by the Ingenuity team • Technical choices, compilation, debug environment
Tuesday January 24th	• 2nd practical work session, assisted by the Ingenuity team • Continuous testing, V&V scripting, live integration
Friday January 27th	• Last practical work session, assisted by the Ingenuity team • Integration and testing with the white board and other agents
Friday February 10th	• Project delivery to n7@ingenuity.io (less than 9MB) • Documentation, ingescape platform for integration and tests, V&V scripts, source code, compiled code

The (multidisciplinary) team involved in this course

Justine

Mathieu



Stéphane

Chloé

Alex

Aurélien

Madeline

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/n7
- The open repository for the **Whiteboard agent**
  - https://gitlab.ingescape.com/learn/whiteboard