**Case study Messaging**

Author: Bartosz Paszkowski

A research and development agency High Sky is working for High Tech companies in the Brainport region. The company has a sound reputation of transforming research prototypes, MVPs, walking skeletons and PoC's into production systems. Recently it was approached to architect a cutting-edge AI project.

The research project involves 15 parties (companies, non-profit organisations and research institutions). The theme of the project is to create a mixed reality in which AI avatars help to train real humans in dangerous or stressful situations. Human is closely monitored by IoT devices and partly by the virtual events in the mixed reality. The system anticipates changes in the mood, stress levels of the trainee and reacts accordingly in real time to provide realistic experiences.

The 15 parties that cooperate on a same project could not agree on the common architecture, tech stack and all have their own specific requirements. The situation is complex because some parties provide software pieces that have been developed for many years now and most of them are written in different programming languages like C/ C++/ python, a subset of .Net (Unity engine) and own twist on C++ (Unreal engine). Note that none of the software is web based.

What was agreed is that each party continues to develop their own technology to keep investment low for each party. The development releases will not be synchronised among parties as it would disturb core business processes of each participant. What was also easy to establish was the data formats and contents that will be exchanged among relevant pieces of software.

The High Sky will need to apply a system of systems engineering to stich all the systems together into one reliable and real-time solution. This is not an easy quest and the most experienced architect, with a good knowledge of [SEBoK](https://www.sebokwiki.org/wiki/Systems_of_Systems_(SoS)) practices is assigned on the job. He worked in the past with enterprise integration patterns, [Ada](https://en.wikipedia.org/wiki/Ada_%28programming_language%29), Erlang and Akka for a demanding military project. This tech stack may not be a best fit for this project but inspires him to think of the following context of this system of systems:

* Emergent properties of the system of systems
* Real systems fail quite often, how to keep the system of systems alive
* Graceful degradation of services
* Distributed system testability and debuggability nightmare
* Real-time properties across system of systems

The architect took his considerations to the technical meeting with all participants. His worst scenario was confirmed. The project needs to:

* React in almost a real-time to the inputs coming from audio, video and wearable sensors analysis services
* Process the sensory data from environment sensing IoT devices and supply it for further analysis to a dedicated service
* An AI component will decide on how the system needs to change
* A mixed reality component will need to react to change requests send by the AI component
* Data hungry AI component needs to learn from all the interactions of all the participants in all the trainings. Data need to be stored across the system for later retraining and simulations.
* There will be no budget for extensive testing, instead it was agreed to record as much as possible of the system behaviour so it can be analysed by eternal tools later.

Your task is to come up with an example system of systems design and a middleware tech stack that fits into the context of this project. The focus of this challenge is to analyse the context of the project and a system of systems design can help you with that. In step 2 select middleware solutions that are almost perfect for the (parts of) the job. Make clear what were your selection criteria and how all alternatives compared on these.