Keeping Borders Safe with Akka

A camera and sensor system with which the Royal Netherlands Marechaussee supports mobile security monitoring near the borders of Germany and Belgium was recently built utilizing components of the Typesafe Platform

About The Royal Netherlands Marechaussee

The Royal Netherlands Marechaussee (Koninklijke Marechaussee; KMAR) is a police organization with military status. It is in many ways a highly visible organization. A holiday flight abroad, for example, often starts with passport control by the KMAR.

One of the tasks the KMAR is charged with is to enforce legislation of aliens. The Schengen Agreement resulted in the disappearance of border control on many of Europe's national borders. To compensate for this, a number of countries took steps to counter illegal immigration and crime. In the Netherlands in May 1994, the KMAR was made responsible for the Mobile Monitoring of Aliens on the internal borders with Belgium and Germany. Mobile supervision operations conducted by the KMAR are intended:

- To combat illegal residence, organized or otherwise (human trafficking or people smuggling) at the earliest possible stage and to prevent and discourage other prospective migrants from travelling to the Netherlands illegally;
- To help combat cross-border crime and migration-related crime.

As part of an initiative to make the supervision operations more effective and rely less on random checks, a high-tech, information driven system was developed to support the activities of the KMAR and improve operational results.

The system is called @MIGO-BORAS – a Dutch/English acronym that stands for Mobile Information-Driven Action – Better Operational Results and Advanced Security.
The Challenge

The new system would have to perform the following three functions:

1. To collect anonymous data for analysis and the construction of traffic profiles;
2. To observe vehicles and select those to be stopped and examined on the basis of analysis;
3. To respond to quick alerts in situations where there has been a serious or large-scale breach of the legal order or public order or in the interests of emergency assistance (as in the case of an amber alert).

For immigration law purposes, anonymous data would only be used for the first two functions (analysis and surveillance). The data stored will not be traceable to individuals since the list of characteristics that forms the basis for data collection and vehicle surveillance and selection is limited. Any data traceable to individuals (such as number plates) will be encrypted before they are processed. In the case of the third application, quick alerts, the data in question is traceable to individual motorists, since in an emergency scenario, one or more specific number plates would be sought.

The Team

CSC took part in a consortium to build part of the @MIGO-BORAS system, the “Sensor Domain” that is comprised of both cameras and sensors that have been installed at the 15 most important border crossings, as well as in all of the cars that patrol the border. There are numerous types of sensors in the domain, and the systems main task is to activate the appropriate combination of sensors and cameras, capture and recognize vehicles in near real time and then route the data to a central system for analysis. Speed is of the essence with this system, as potential suspects need to be identified and stopped close to the border by an officer of the KMAR on location in the case of quick alerts.

Which Technology?

One of the hardest problems to solve was that of aggregating the real-time raw data of sensors and software processes into a single view of a vehicle passing the border; for example radars detect a vehicle, cameras take photos of the vehicle, while other processes detect the length and color of a vehicle. None of these sensors know about each other yet the system needs to assess the situation, choreograph the receipt of messages from multiple sensor and camera arrays, at the same time the system cannot wait for one signal to process before it processes the next – the wait would be too long. Many vehicle properties need to be automatically recognized by the system and consolidated into one view of the vehicle at high precision, to be sent in time for officers to respond to as quick as possible.
Many vehicles, many lanes, fixed sensor arrays and patrol cars, many servers and it all has to happen at the same time.

Raymond Roestenburg, the CSC Architect involved in the project was charged with looking for new technologies that would help him build this high performance, highly distributed event-driven system.

While Raymond had built similar systems before, the sheer magnitude of this system, with numerous border checkpoints, distributed sensor arrays both across the border and in patrol cars meant building a system that was highly asynchronous and concurrent and clearly required new technologies not previously used. He initially looked at utilizing Java, Google protobuf and JBoss Netty however the technologies, while viable were far too low-level and would have required too much development effort to implement such a large system.

**The Solution**

Raymond then found Typesafe’s Akka, a toolkit and runtime for building highly concurrent, distributed, and fault tolerant event-driven applications on the JVM that leveraged all of these technologies and provided a higher level of abstraction with it’s Actor based model. The first thing that drew Raymond to Akka was how easy it was to work with Akka Actors, subsequently simplifying the problem of processing the raw sensor data and aggregating it to one vehicle view asynchronously. Furthermore, the cellular technology utilized for data transmission required that a balance be struck between available speed and bandwidth, which was easy to do in the Actor model as it’s designed with lightweight protocols in mind.

Another aspect of leveraging Akka was the ability to utilize Java or Scala as the programming language. Raymond had long considered utilizing Scala over Java as it is far less verbose and given the tight timeframes of the project this was a logical decision. Typesafe’s Scala is a general purpose programming language designed to express common programming patterns in a concise, elegant, and type-safe way. It smoothly integrates features of object-oriented and functional languages, enabling Java and other programmers to be more productive. Code sizes are typically reduced by a factor of two to three when compared to an equivalent Java application.

**The Deliverable**

The stakes for implementing a nationwide system against tightly specified requirements were very high.

Ultimately, a team of three developers produced a fault tolerant, highly reliable Sensor Domain in a fraction of the time that it would have taken with other technologies.

Akka is running on both industrial servers (a fit for purpose server that can operate in a higher range of temperature, vibrations and other environmental aspects) in the patrol cars, on sensor arrays, and on a central clustered system in a data center that coordinates all the sensors and data.

Not only does the system aggregate the raw data of the sensors, but it also does the processing as well (license plate, country recognition, vehicle classification) which is an extremely CPU intensive task that is executed asynchronously via Akka actors.

The recognition is done with independent image recognition software libraries to obtain better confidence results, which subsequently have to be joined to one result (asynchronously as one can take longer than the other), which was very easy with Akka. Similarly, the filtering of “double triggers” (the same car detected
by many cameras), where the client only wants one registration of a vehicle was easy to implement in Akka. Scala its ability to seamlessly integrate with Java made it possible to reuse existing libraries for image processing.

Raymond's team felt that Akka's quality was very high compared to other open source technologies that they have dealt with. It's extremely stable, and the issues that they did discover were addressed and fixed rapidly.

On August 1st, 2012 the Dutch Minister of Defense made it official. It has been covered extensively in the news media and is achieving its very high performance requirements.

The **Typesafe Platform** is a modern software platform that makes it easy for developers to build scalable software applications. It combines **Play Framework**, **Akka** runtime, the **Scala** programming language, and robust developer tools in a simple package that integrates seamlessly with existing Java infrastructure. Commercial support and maintenance is available for the Typesafe Platform through the **Typesafe Subscription**.