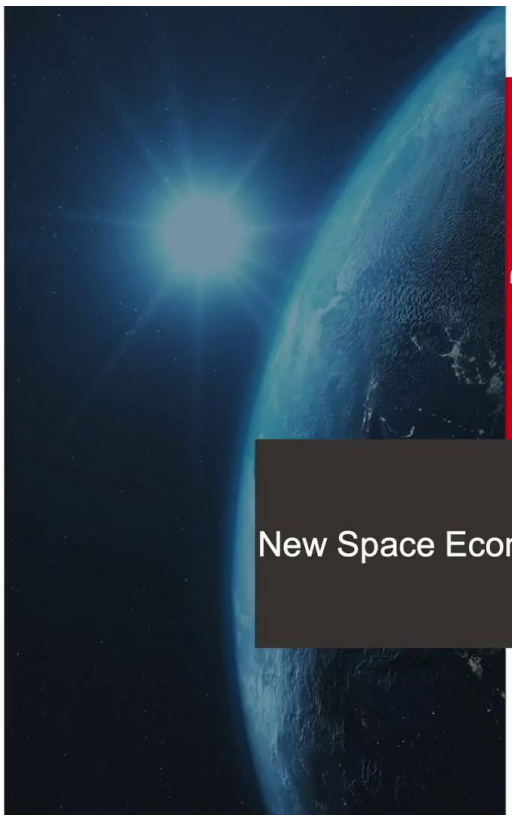
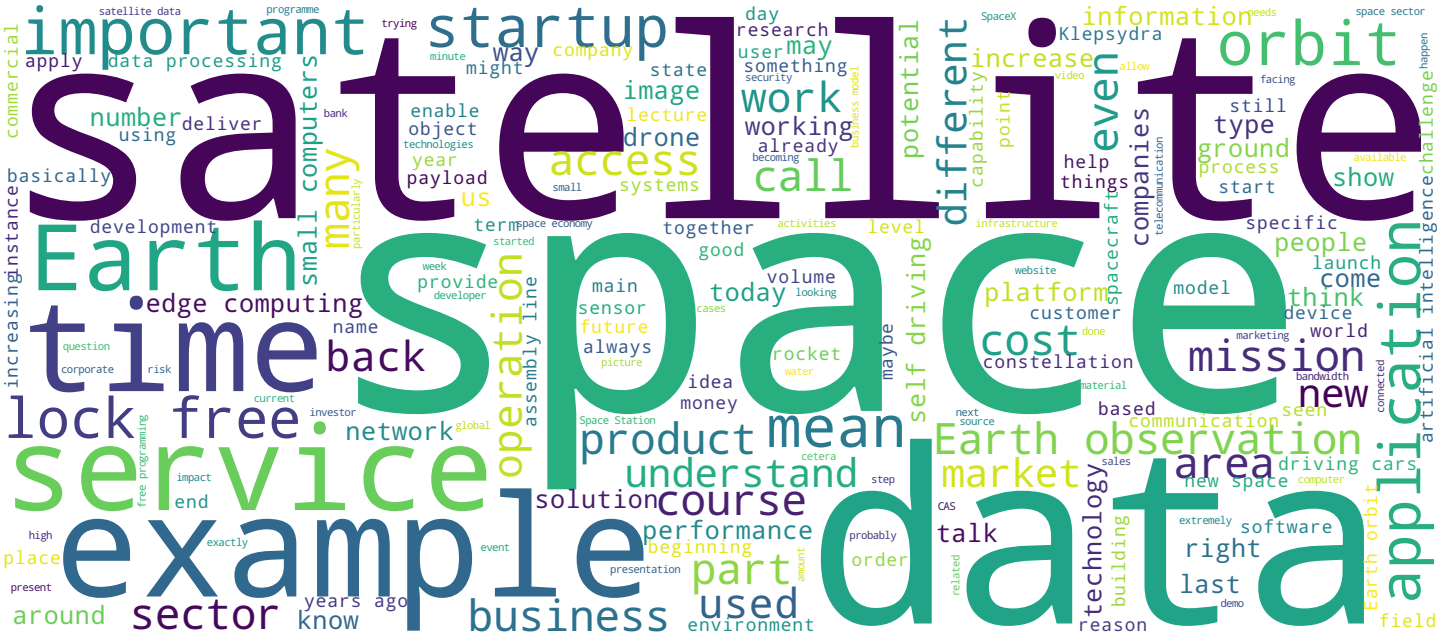


# Klepsydra's ring-buffer: "There and back again"

## New Space Economy

Dr Pablo Ghiglino, Klepsydra Technologies




EPFL

# Short Bio

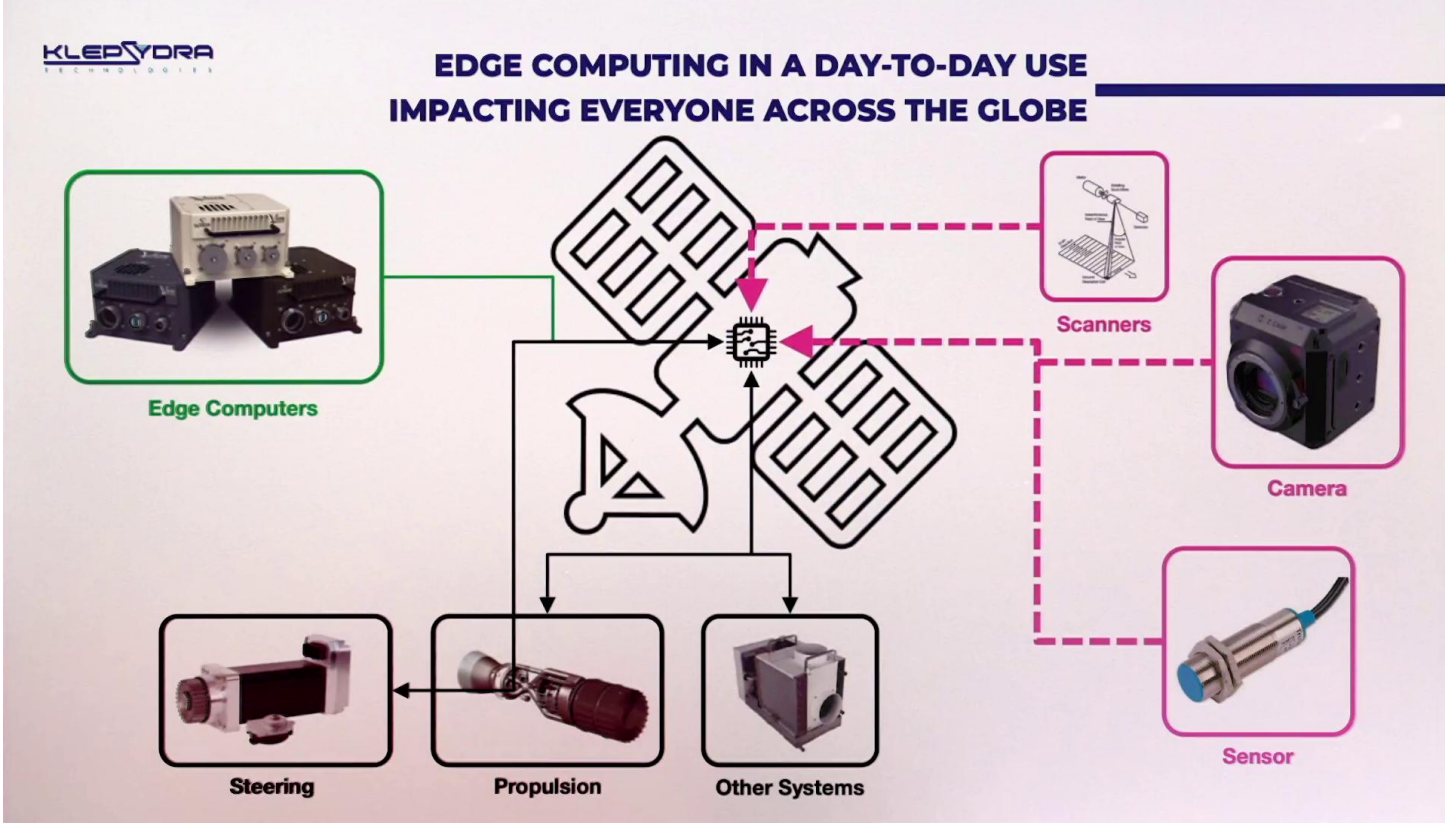
- Studies
  - BSc Electronic Engineering, UPM, Spain
  - MSc Electronic Engineering, Uni-Stuttgart
  - PhD Aerospace Eng., Surrey Space Center
- Work Experience
  - 30+ years programming experience
  - 12 years developing trading system
  - 3 years founder of Klepsydra Technologies
- Klepsydra:
  - Combination of previous experiences, applied to edge computing.

Hello everyone, my name is Pablo Ghiglino. I am the founder of Klepsydra Technologies. And I'm here to present the work we have been doing in Klepsydra in the area of data processing, something we call the ring-buffer. We call the presentation, There and Back Again as a tribute to the beloved Hobbit book. And the reason for this naming is that what we have developed comes from Earth's applications. We deploy it to space, and now we are finding new applications on Earth through this technology back again. So this is exactly the reason why they call the presentation like this. A short bio about myself. I study Electric and Electronic Engineering and I have a PhD in Aerospace Engineering. I have been working in IT or programming. I've been a developer for more than 30 years, since I was 12 years old. But more importantly, the last 12 years, I work a lot in an area called trading system, which I will explain in a minute. And for the last three years, I have founded my own company, Klepsydra Technologies, what I am working on at the moment. Klepsydra, which I will present in a minute, is a combination of all these previous experience in aerospace and also as a developer of trading system, and apply to something called edge computing.

Notes

Summary

0m 05s

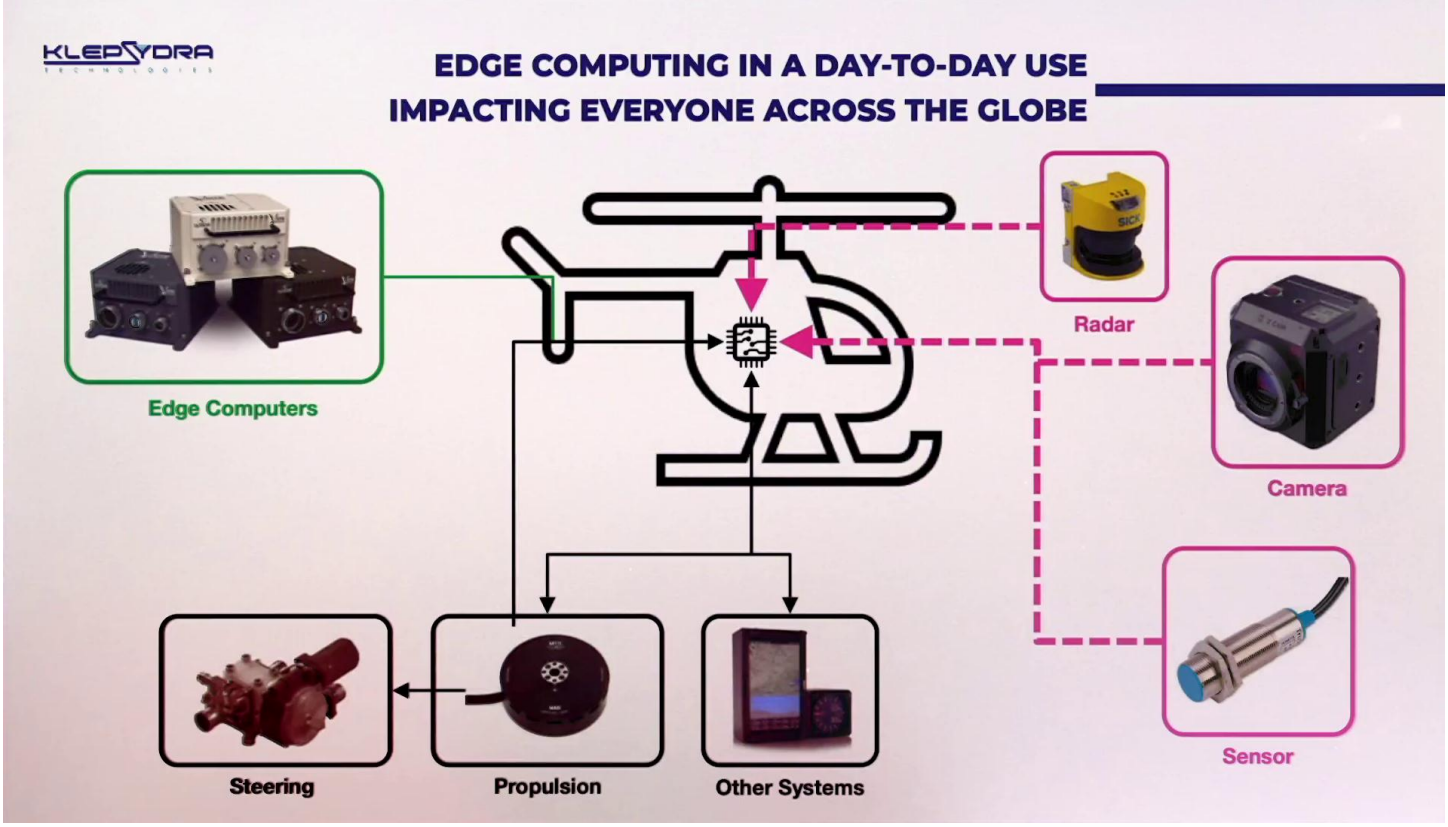



So what is edge computing? Edge computing refers to these small computers installed inside different type of devices. In this case, we're talking about satellite. The key feature of these small computers installed in these devices is they're connected to a large number of sensors and other sources of data. The newer scenarios, the newer computers tend to be connected to more and more data sources. New satellites produce volumes of data that had not been seen before and these small computers have to process it all. This is the case of space. However, the same happens in other sectors.

Notes

Summary

1m 21s

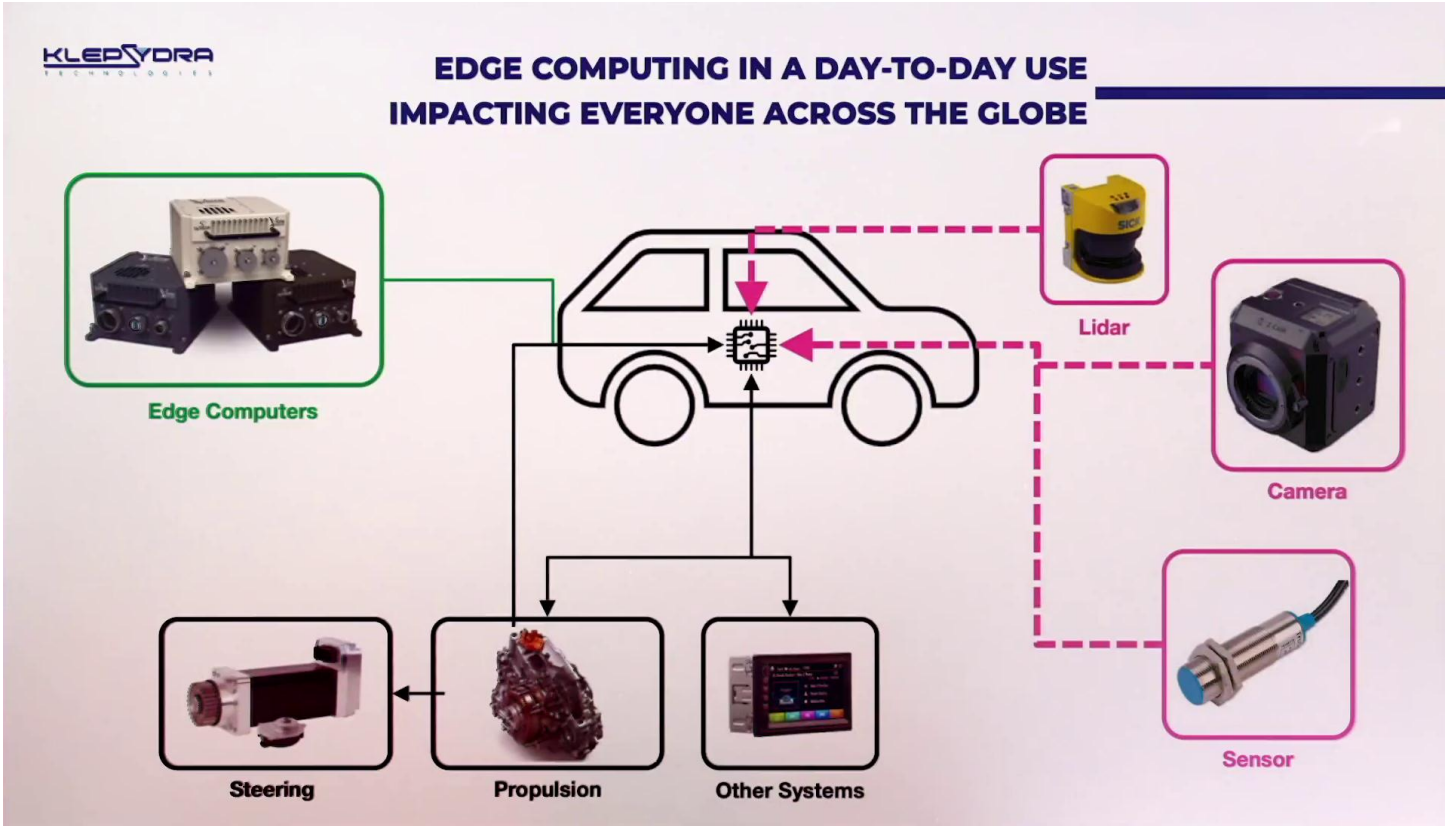



For example, in aerospace, helicopters, aircrafts, and drones are facing exactly the same situation. They have small computers inside connected to a growing number of data-producing element, sensors, networks, et cetera.

Notes

Summary

2m 03s

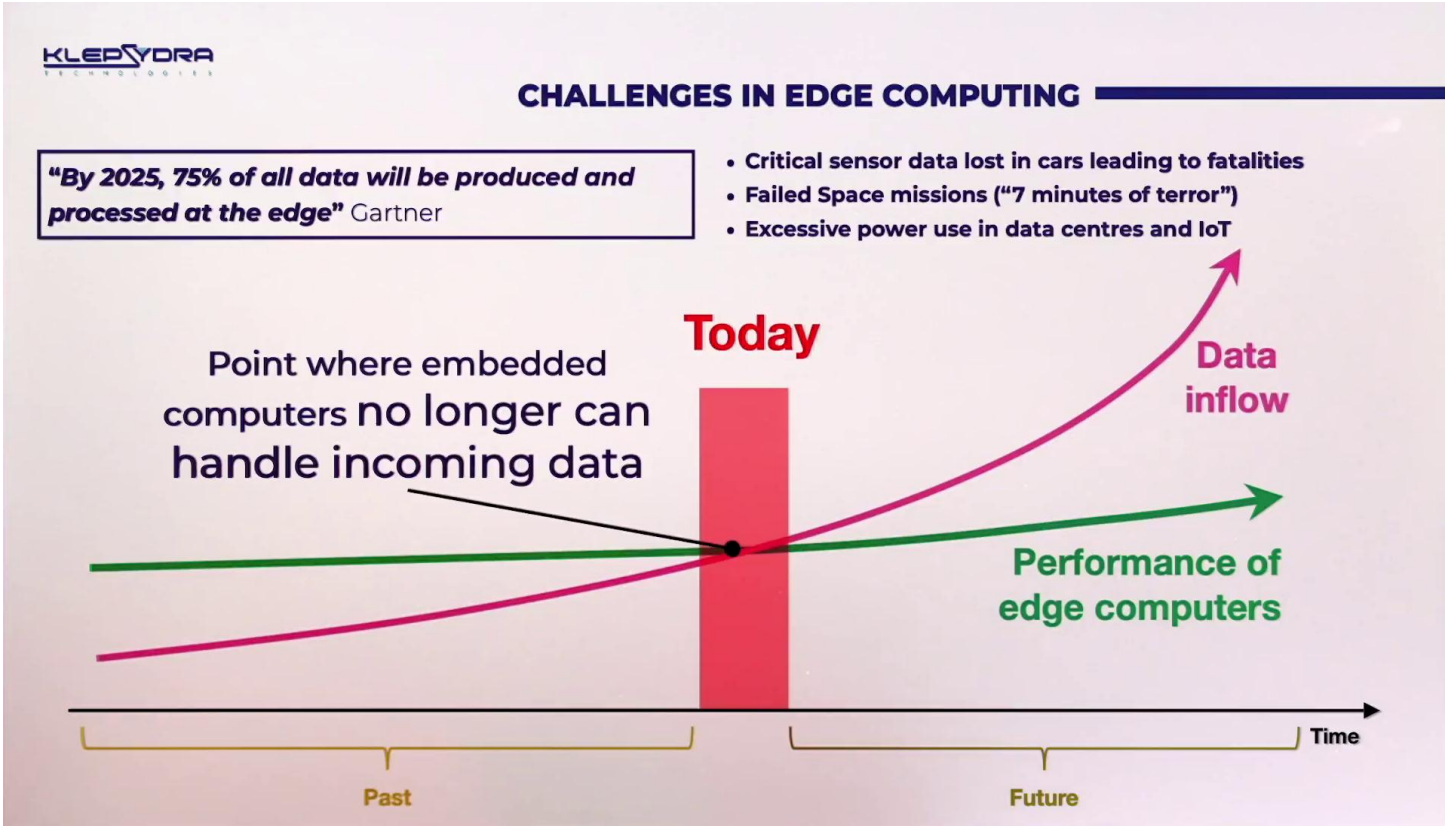



Again, the same happened to another sector which is particularly important because it's where the number of sources of data has grown the most, which is automotive. Nowadays, a car can have up to 100 sensors or even more. Self-driving cars are facing the use of even better sensors and even more sensors, and they are very much stretching the capabilities of these small computers. And this is one of the reasons why we haven't seen self-driving cars yet in the streets. The problem of data processing is absolutely massive in this sector.

Notes

Summary

2m 18s

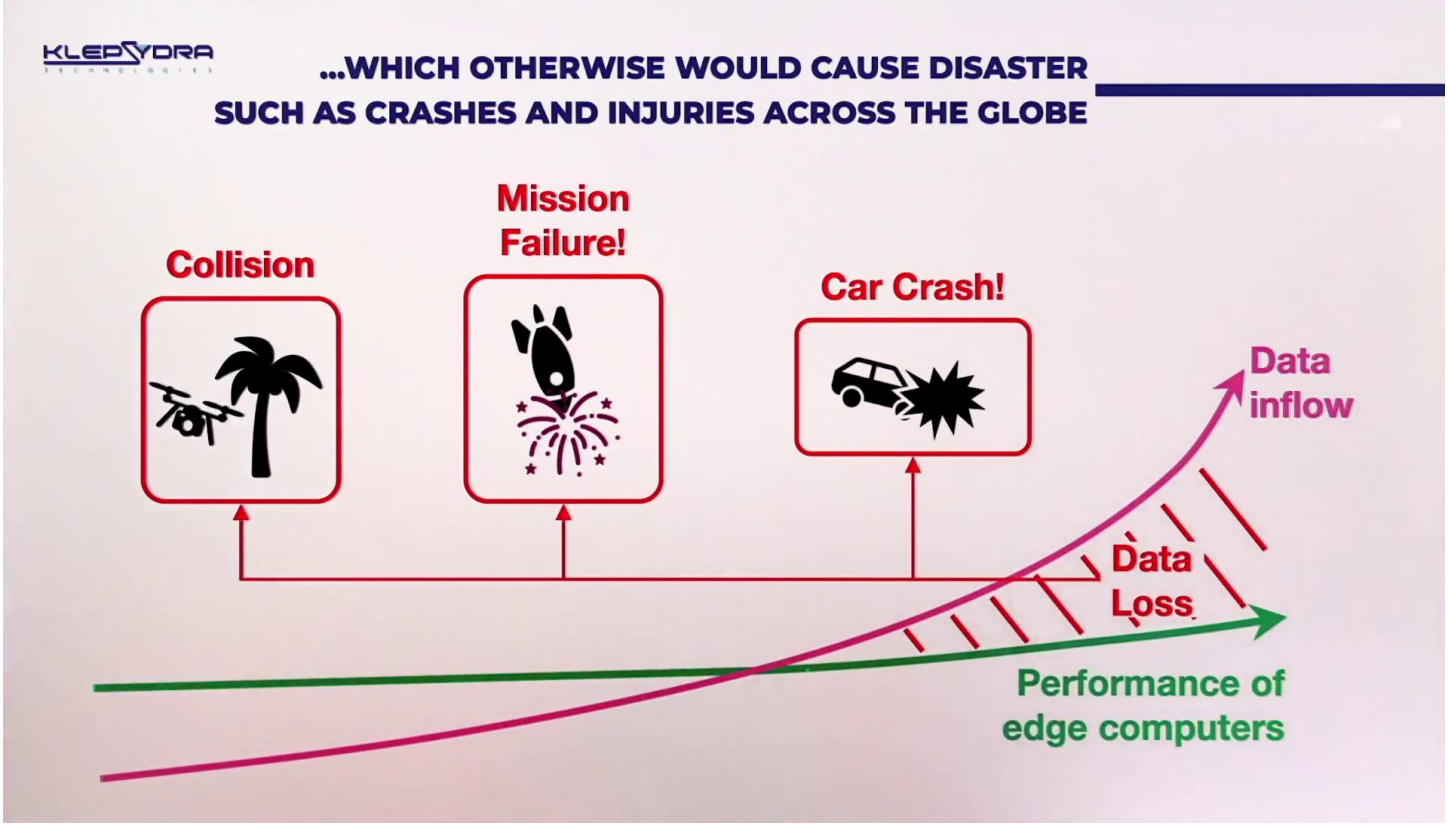



In particular, edge computing, which is what I mentioned, these small computers, are facing a very important issue today, which is that while the computers are increasing as well, increasing in capability, there are obviously better computers for the edge, better and better every year, that's for sure. However, the volume of data that this computer has to process grows even faster. So we are starting to see a gap between the volume of data and the capability of the computer.

Notes

Summary

2m 56s

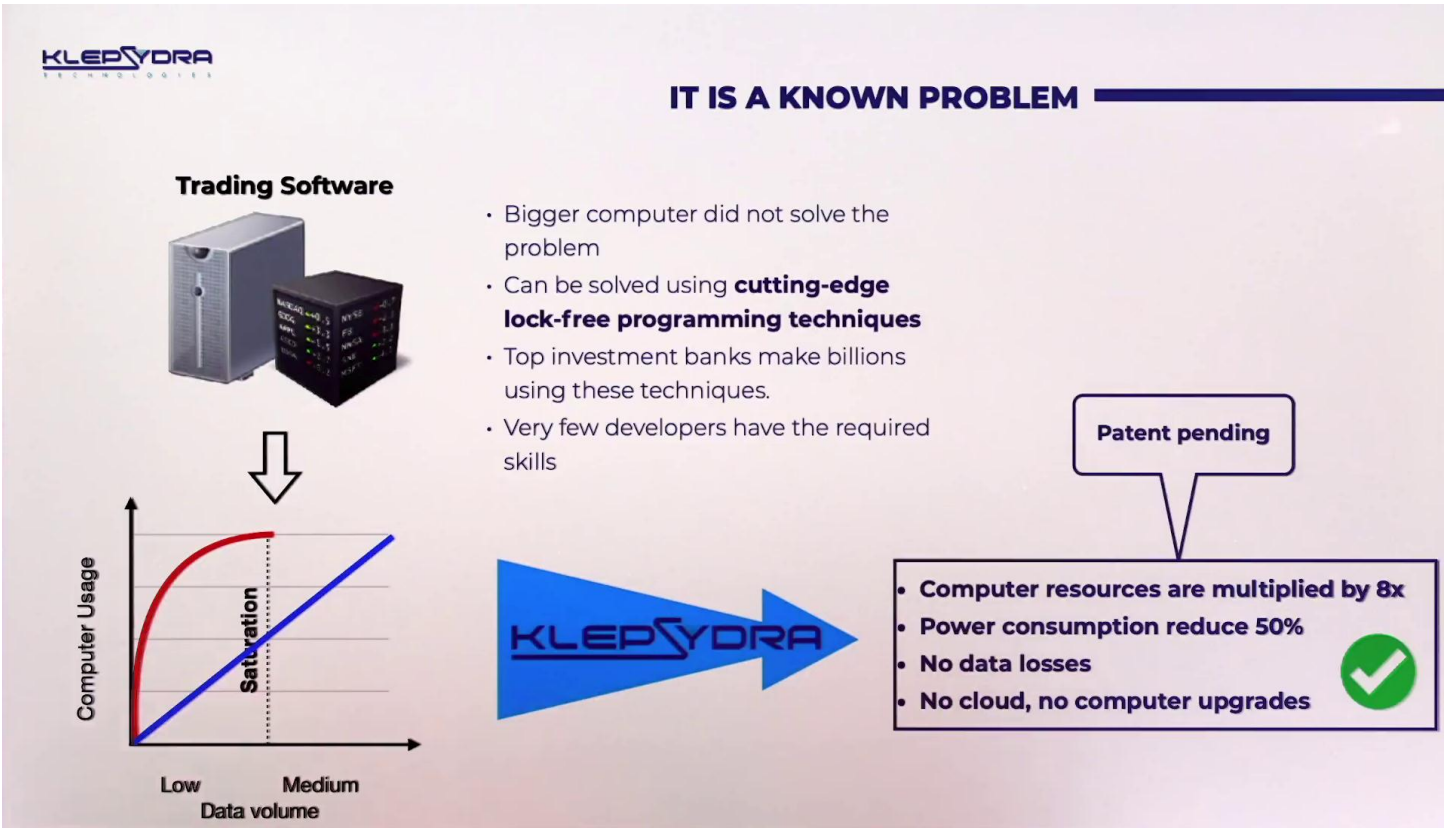



Why is this important? Because this area here, this gap, corresponds to data losses. This is where the data is lost. We don't process some of the data arriving to the computer, and this produce all sorts of problems. Collisions, in the case of robotics and drones. More importantly, mission failures in the case of space. Many past famous mission failures are because of this data-processing issue. And like I said earlier, the most important of all is the self-driving cars, what is preventing the appearance of the real self-driving car in a production environment.

Notes

Summary

3m 26s

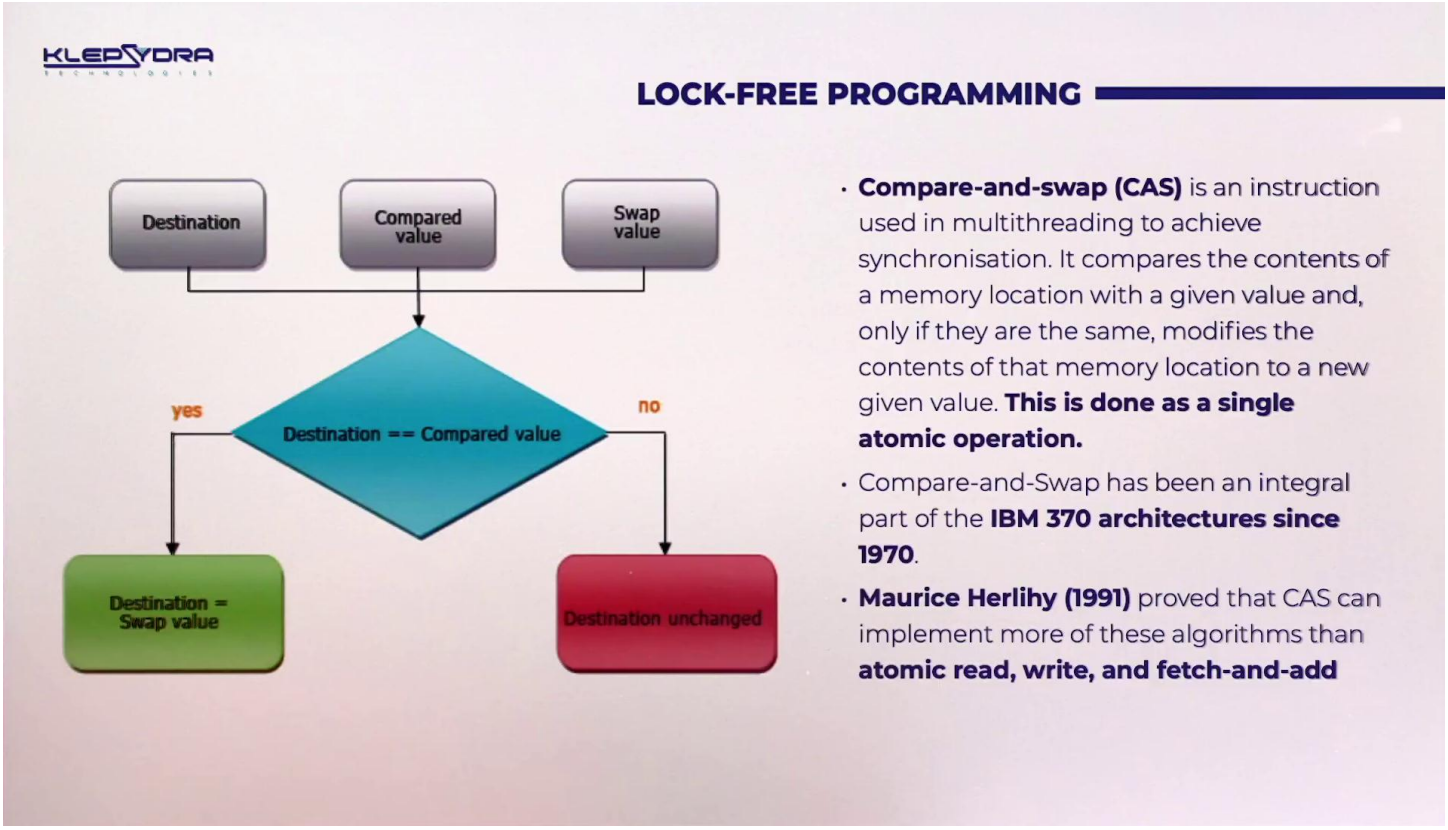



This problem, however, is not new. It's not a problem that has been discovered now with edge computing, it has existed for a while. It appears roughly 10 years ago in this sector that I call out at the beginning called trading. Trading is a technique used by banks to perform transactions against a stock exchange. The more transaction they prefer per day, the more money they make. This is pretty much in essence what banks are doing in terms of trading software. And like I said, 10 years ago, there was a big revolution because there was a development of certain techniques called lock-free programming. These algorithms using lock-free programming are extremely fast, can increase the performance of a computer substantially up to four times, and in some cases, up to 10 times. So what I have done in Klepsydra initially was to use the same techniques but apply to edge computers. So we use exactly the same techniques as trading systems, but apply to edge computing, achieving this level of performance as you can see here.

Notes

Summary

4m 01s




So what are these lock-free programming techniques? Lock-free, the technical name is this, compare-and-swap. Compare-and-swap refers to computer instruction in the processor that enables you to update a piece of data in parallel by multiple threads, by multiple elements, trying to update the same data at the same time. And this can be achieved by using compare-and-swap. Okay, this is the most important element of this. It's an atomic operation in the sense that this is just one operation, not many. It's one operation. This is why the name is like this. Moreover, these techniques were invented roughly by the end of 70s, so it's 50 years ago, but they didn't really become popular until the early 90s when these were made available to the most high-level languages like Java and C++. But the real revolution came 20 years later, like I said, when the appearance of high-frequency trading techniques in the banks.

Notes

Summary

5m 06s

KLEPSYDRA TECHNOLOGIES

## LOCK-FREE PROGRAMMING

- Threads need to acquire lock to access resource.
- Context switch:
  - Suspended while resource is locked by someone else
  - Awaken when resource is available.
- Not deterministic, power consuming context switch.

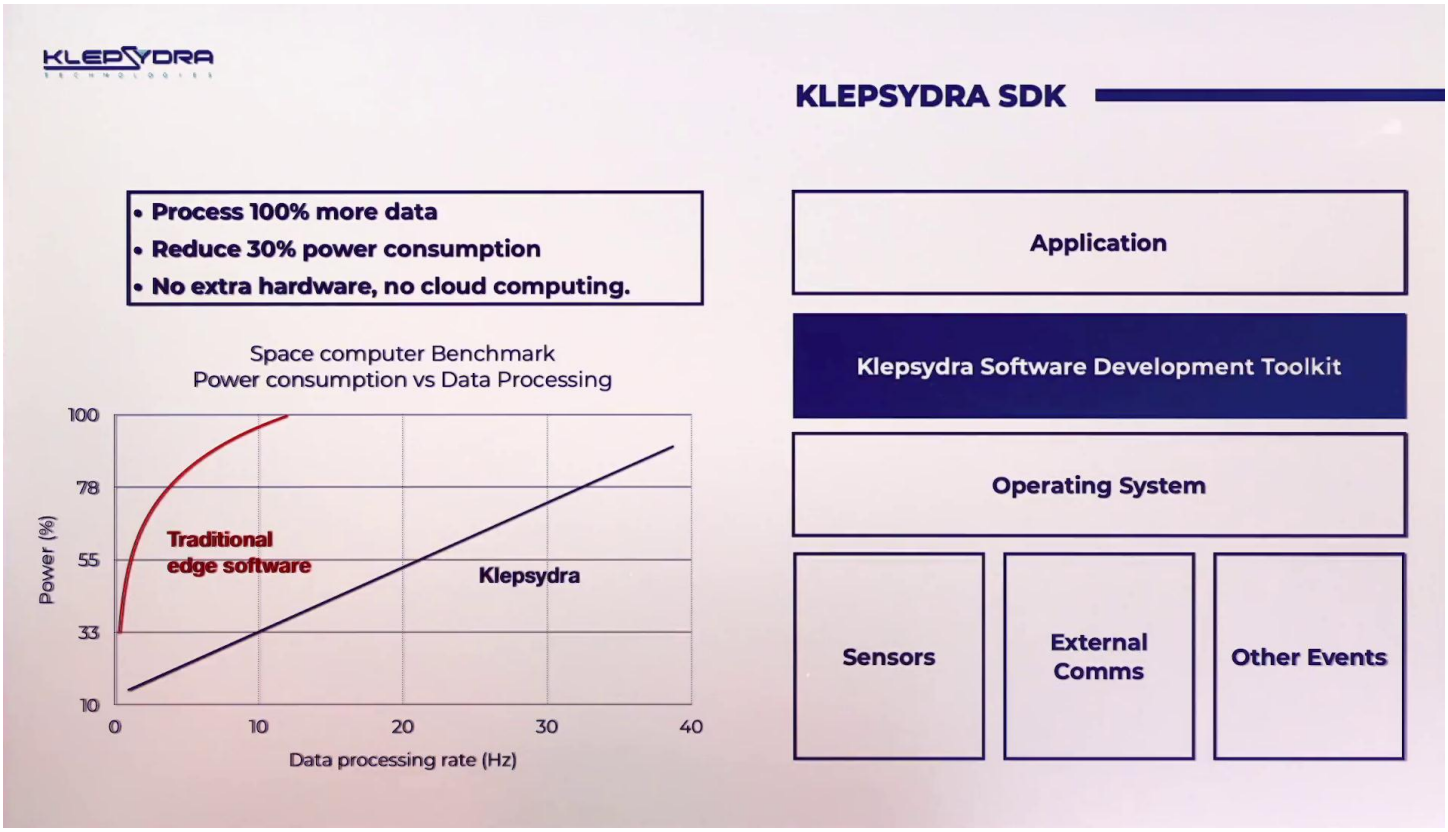
- Threads access resources using 'Atomic Operations'
- Compare and Swap (CAS):
  - Try to update a memory entry
  - If not possible tried again
  - No locks involved, but 'busy wait'
- No context switch required.

So why is this lock-free particularly fast? Why these CAS operations are fast? Without CAS, before the existence of CAS, the world of parallel processing in a computer looked like this. A lot of people cramming, trying to access resources, fighting each other, trying to access to a place, or something like this. And this was extremely power-consuming, non-deterministic, so that means that you don't know what is going to happen in the next millisecond of the computer. And these are very heavy operations. This is why at the moment, many of these sectors that I mentioned, including space, are not allowed to use these parallelization techniques. However, when we talk about CAS, the behaviour is completely different. You have to think of CAS as opposed to this chaotic queue, as a luggage belt, an orderly luggage belt where people can just go and collect their luggage. You cannot access the whole belt, you can access just your luggage. This is the most important thing. But you always have access to it in an orderly manner. This is a more granular way of allocating resources, more or less. This is why this is particularly fast, particularly low wait, and are becoming a very important technique in the parallelization of data processing.

Notes

Summary

6m 05s

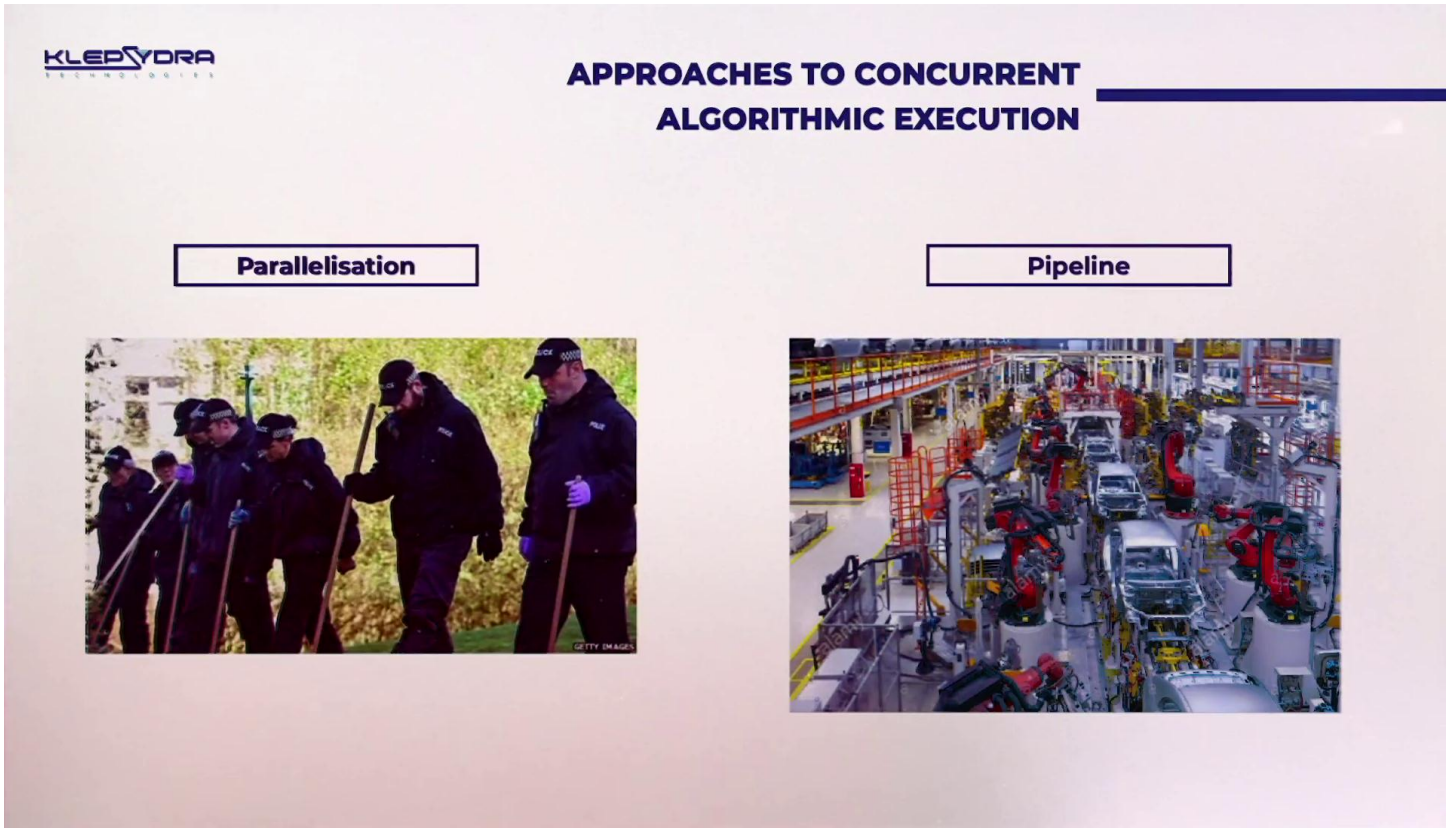



At Klepsydra, we have built a software for this, a solution. We call it SDK. It's basically a framework that developers can use to develop software using this. Just to show you the level of performance we have, here is a chart for a space computer. We see a similar chart to what I showed you at the beginning in the part of trading where Klepsydra can increase, in this case, four times the volume of data that can be processed in these computers. Just by using these techniques coming from trading.

Notes

Summary

7m 27s

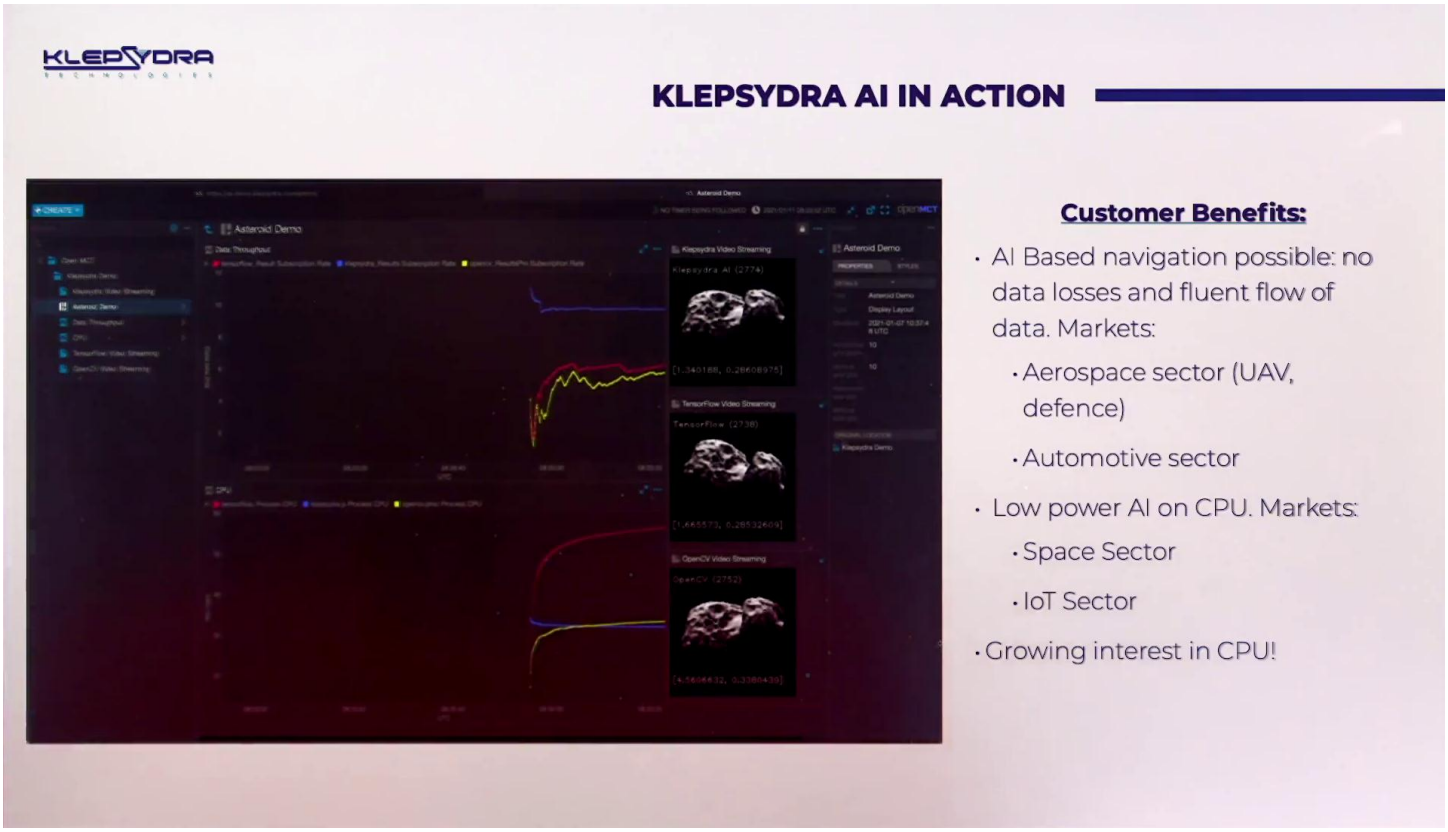



Moreover, we have applied another technique. I'll invite you to look into researching this field called pipelining and parallelization. I really much think that these areas are of massive growth, and for those of you that are studying, it is a very important area, as well as lock-free. The traditional parallelization techniques, or vectorization as they're also known, enable you to perform large mathematical operations in a parallel manner. So for example, if you are performing a matrix multiplication, instead of multiplying different parts altogether, you multiply different parts in different threads in parallel. However, pipeline is a different technique in which you have behaviour similar to those in an assembly line, where each part of the assembly line perform one task. This technique is particularly well-suited for these scenarios because it reduces the power consumption and increase the amount of data that can be processed. Imagine that each car in this assembly line is like an image, for example, coming from a camera, and you want to do certain things with the image, object detection, filtering, et cetera. So each step of the assembly line will be performing one part of this algorithm on the image.

Notes

Summary

7m 54s

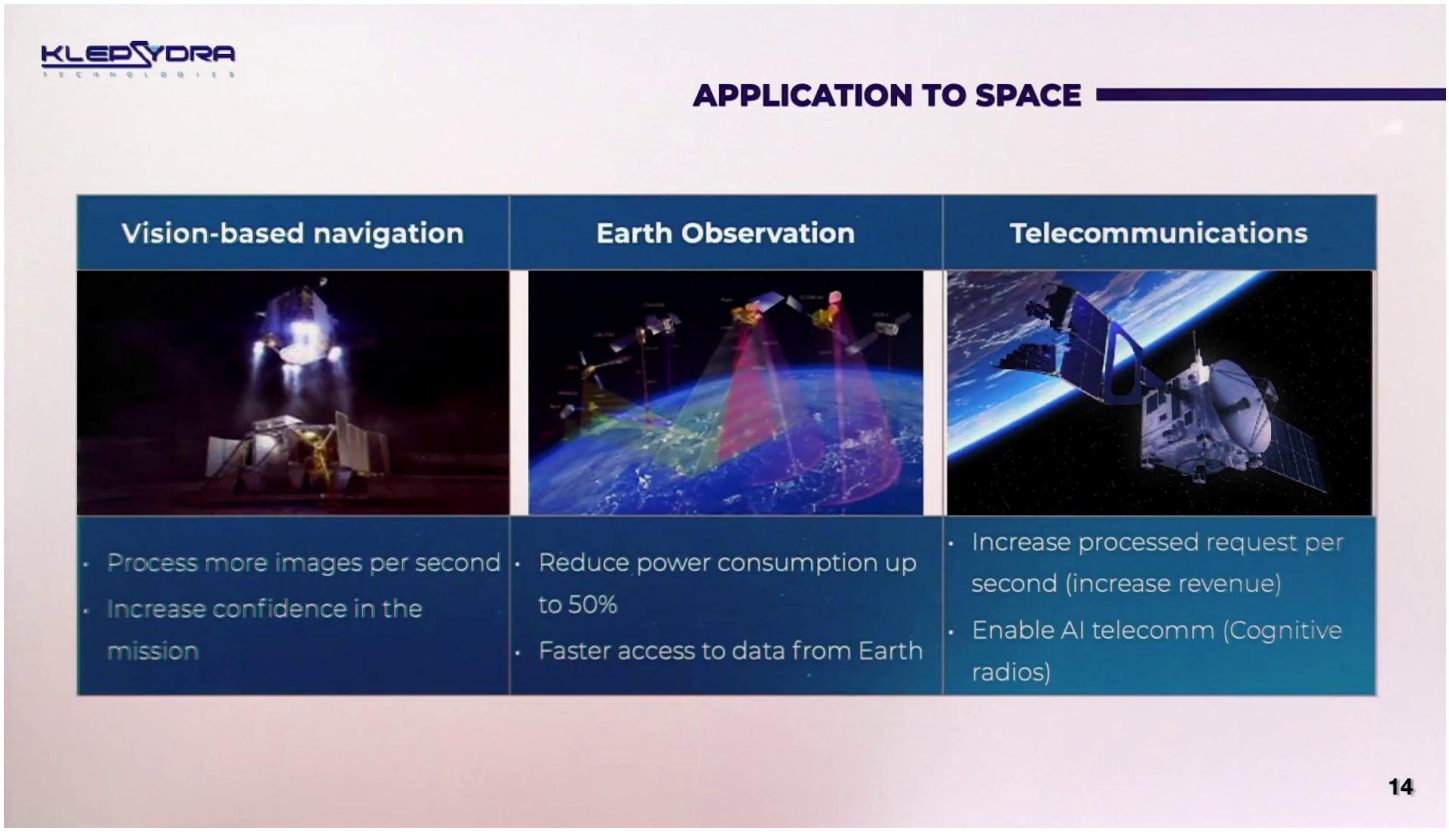



With this, we have developed yet another product combining our lock-free technique and the pipelining, which is our artificial intelligence software. There is a demo in our website. If you go to the website, you will see this demo that shows the level of performance we can achieve. The demo is for an area called vision navigation. In particular, we are analysing images from an asteroid and trying to detect how to land on it using artificial intelligence. And then we have the chart. Throughput, so how much data we're actually processing. Blue line is Klepsydra. As you can see, we process double than traditional techniques. And CPU usage, where Klepsydra, the blue line again, is using half of the power as one of the market leaders in the field. Why is this important? Because we can process double amount of data with half of the power, meaning that this can be placed in this small satellite that I show at the beginning, and enable it to perform navigations without any incidents as opposed to the current situation where this is a very risky manoeuvre.

Notes

Summary

9m 22s

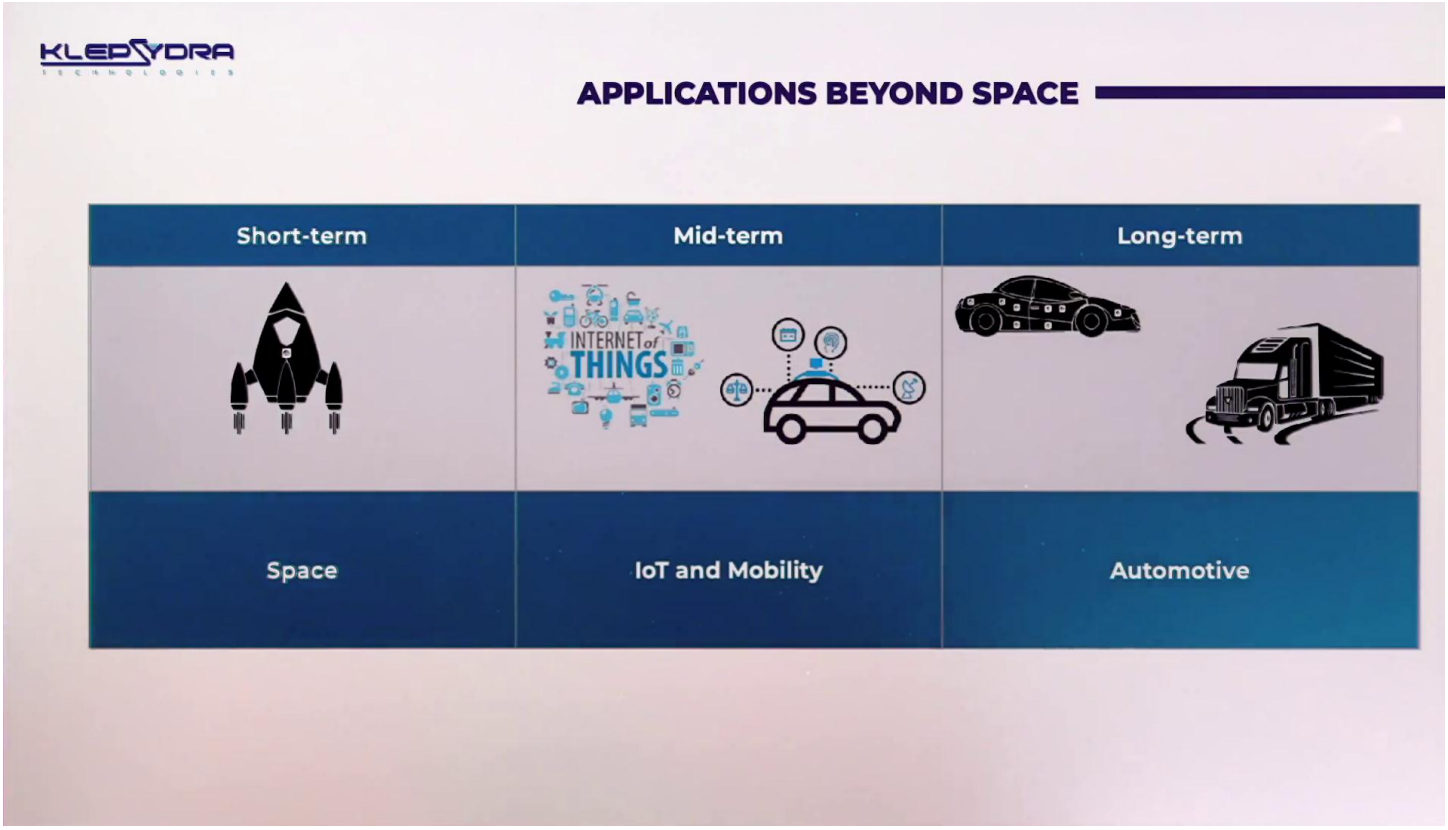



The application for space are not only related to vision, like I said, vision navigation, but also to Earth observation which refers to satellites with cameras pointing to Earth and analysing the data, but also for telecommunication satellites. This is another area where artificial intelligence is becoming very important. And this is the journey up of The Hobbit.

Notes

Summary

10m 22s




Now, the back-again part is that we are seeing that there is very good applications as well on Earth, from IoT, robotics, et cetera, but more importantly, what I said at the beginning about self-driving cars. Self-driving cars are really facing this massive issue of data processing with massive data losses, not only a software like the one I presented coming from the trading techniques can really solve.

Notes

Summary

10m 43s

# Conclusions

- Investment banks invested the fastest software in the world for High Frequency Trading.
- Klepsydra software enables this technology to the Space sector.
- Other terrestrial sectors (robotics, automotive, etc) can also benefit from this technology

Thank you!

So in conclusion, what I showed you here is a technique coming from a terrestrial application, from investment banking. It's a software that is extremely fast. It's probably one of the fastest software techniques in the world. We have enabled this software to work well in the space sector, and now we're seeing that this software can be used in other sectors as well. So there and back again, okay? I invite you again to look into this research that I mentioned, pipelining and lock-free. And I thank you very much for your attention. Bye bye.

Notes

Summary

11m 09s