

Fighting Climate Change by Building Sustainable Software

Ioannis Kolaxis – Software Architect / Distinguished Expert

GOTO Amsterdam 2022

Short Biography



Ioannis Kolaxis **Architect / Distinguished Expert**

- Book author: **Green Software**
- **Inventor** (5 Patents)
- **1st winner:** Atos **Innovation** Week 2021 & 2020
- Worked for IBM, SIEMENS
- **Speaker** (Oracle Code One 2019)



@IoannisKolaxis



kolaxis.dev

You Can Fight Climate Change!

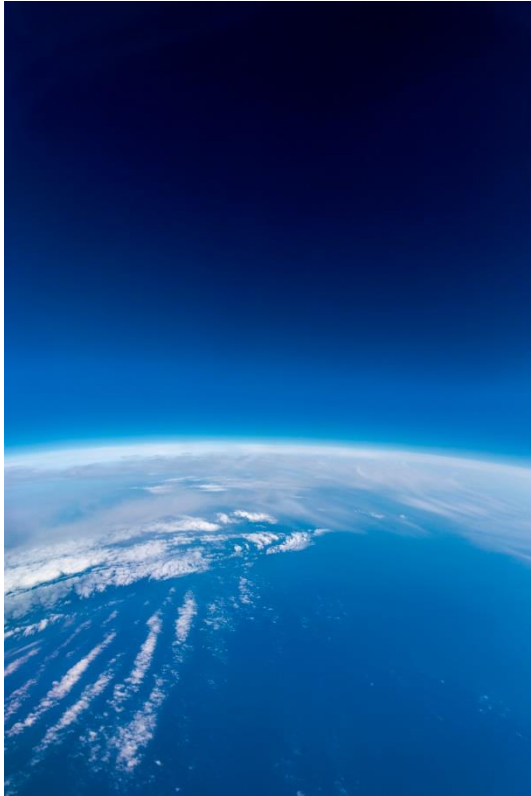
By Following Sustainable Practices in Your Work



By the end of this talk, you will be able to make a **real impact** in your work as **IT professional**, to help prevent **climate change**

Climate Change

What is the Problem?



Greenhouse gases (CO_2 , CH_4 , ...) in earth's atmosphere trap **heat**
→ **increasing temperature**

Who emits the most carbon?

a. Industry

b. Transport

c. Electricity &
Heat Producers



CO₂ Emissions Per Sector (2018)



Electricity &
Heat Producers

13.978 Mt CO₂

42%

Transport

8.258 Mt CO₂

25%

Industry

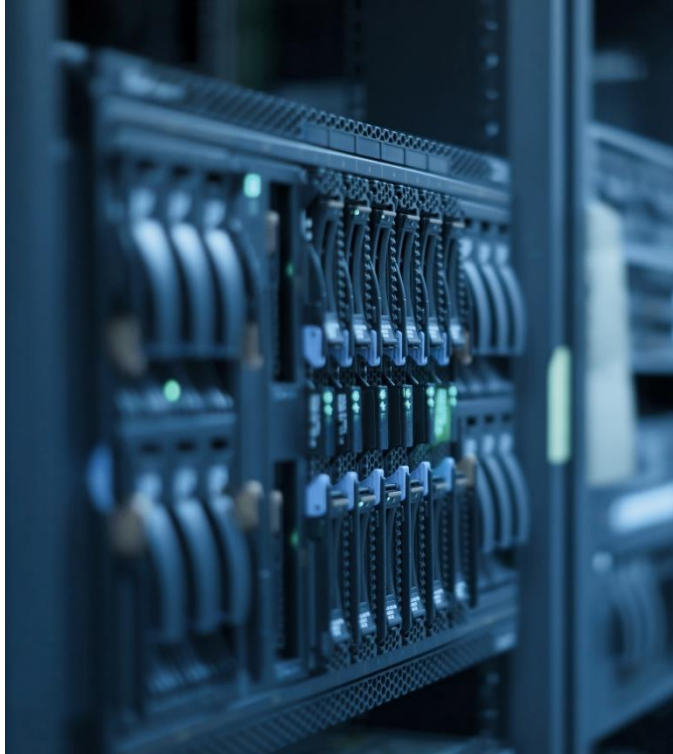
6.158 Mt CO₂

18%

Source: IEA, Data & Statistics <https://www.iea.org/>

Our Contribution To Climate Change

Electricity Consumed by IT

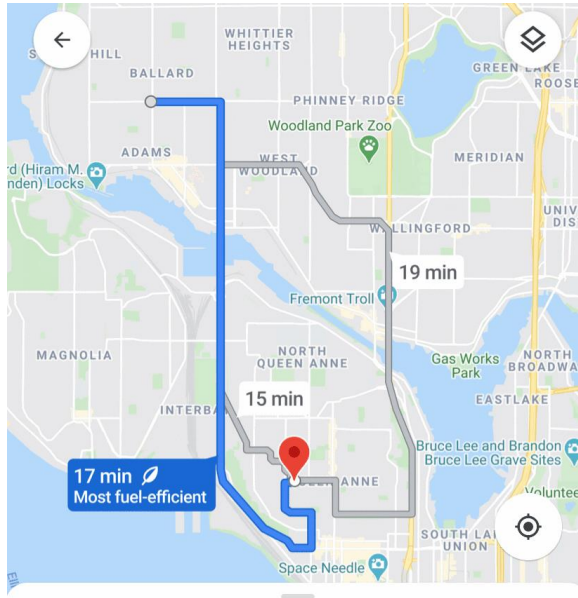


~**2%** of global electricity in 2019
was consumed by:

- ▶ **Data centers** (200 TWh)
- ▶ **Data networks** (250 TWh)

Source: IEA (2020), Data Centres and Data Transmission Networks, IEA, Paris
<https://www.iea.org/reports/data-centres-and-data-transmission-networks>

Finding the Best Route



🚗 Drive

17 min (4.8 mi)

Most fuel-efficient route, the usual traffic



8% lower CO2 emissions than the fastest route
Based on average fuel consumption for vehicles in
your region. [Learn more](#)

► Options:

- Fastest route
- Shortest distance
- Fuel-efficient route

Source: <https://blog.google/products/maps/redefining-what-map-can-be-new-information-and-ai/>

Data Networks

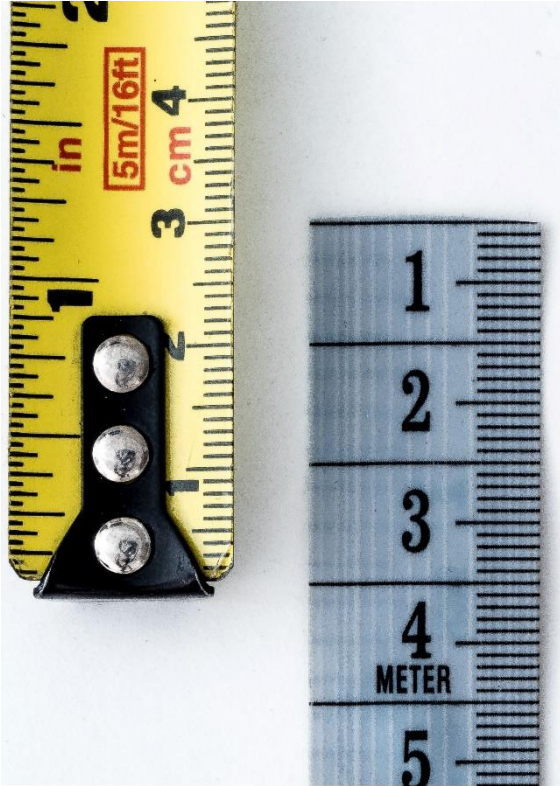
How can we consume less electricity?



- ▶ Minimize distance (hops) for data transferred over the network
 - Place your servers close to your customers
 - Process data at its source (Edge)

Data Networks

How can we consume less electricity?



- ▶ Reduce **size of data** sent over the network
 - Filter out unnecessary data
 - Compress before transfer

Journey to Decarbonization



Don't send your **data**
around the world!

Data Centers

How can we consume greener electricity?



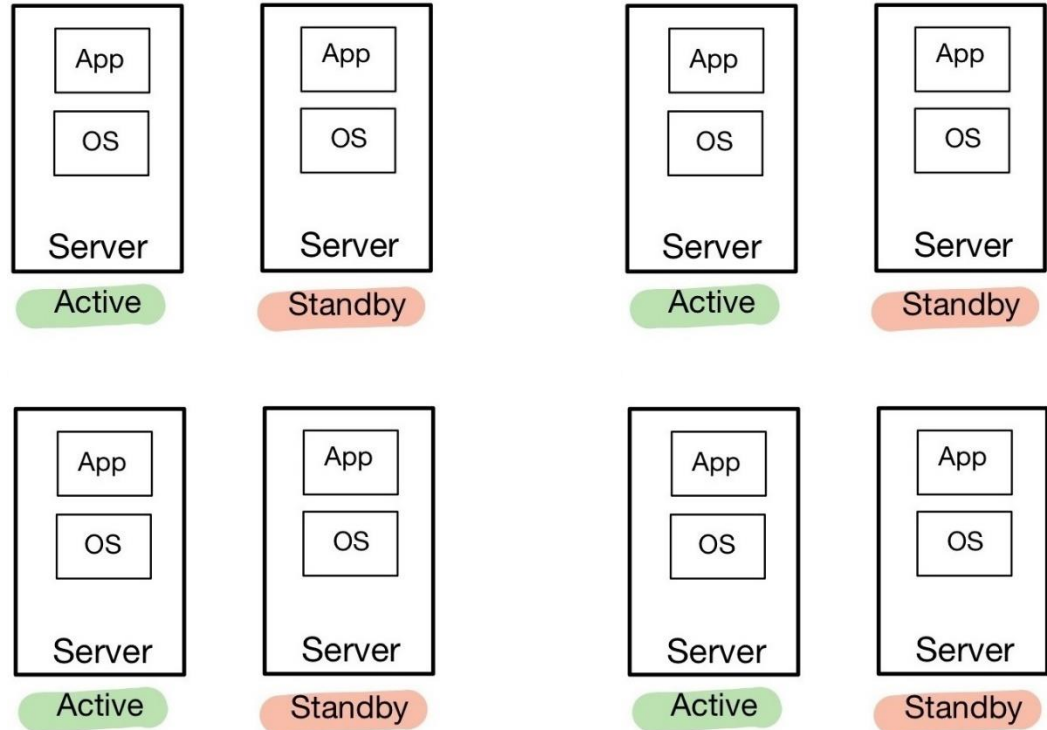
- ▶ Host your application in a region that consumes clean energy
 - Europe: Finland, Belgium, Zurich
 - North America: Montréal
 - South America: São Paulo
 - USA: Oregon, Iowa

Source: <https://cloud.google.com/sustainability/region-carbon>

Up to **50%** of the servers in your data center are idle, doing nothing: they just **waste energy!**

Data Centers

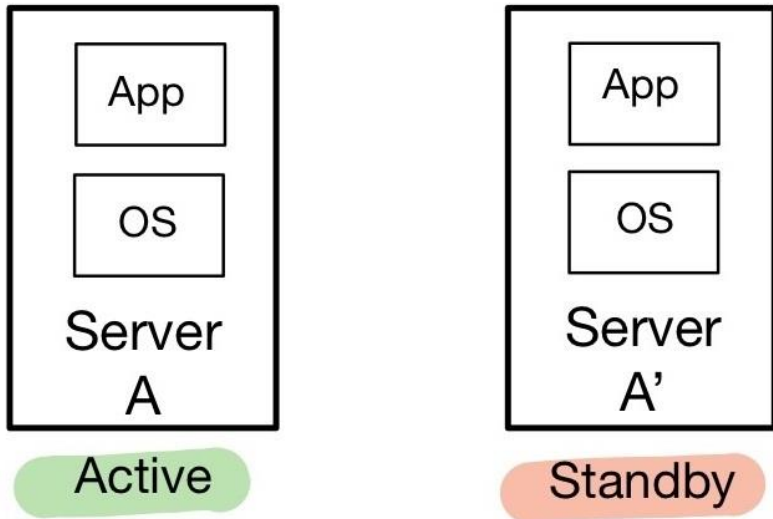
How can we consume **less** electricity?



Using Redundant Servers

For Highly-Available Applications

Normal Operation



Server A handles every request to the application

Server A' is redundant, not serving any requests, just waiting for a failure to happen

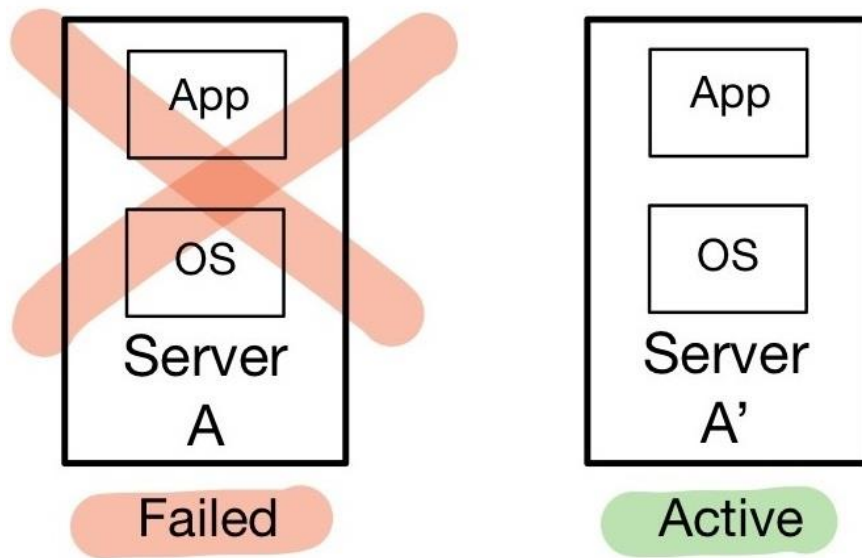
Using Redundant Servers

For Highly-Available Applications

Server A had failed!

Server A' (previously standby/redundant) now handles every request to the application

Failover



A Car With Redundant Engine



- ▶ Citroën 2CV Sahara (1958-1966)
 - 2 engines for redundancy
 - 2 petrol tanks

Containers can be started in a fraction of the time it takes for **Virtual Machines**

Redundant Servers

Why Do We Need Them?

- ▶ If we didn't have a redundant server ...
 - How long would it take us to spin up a new instance?

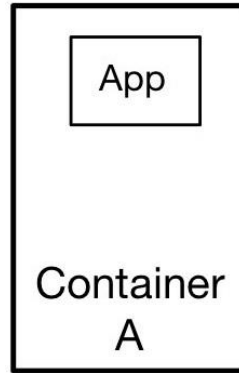
	Virtual Machines	Containers
Size	GB	MB
Startup time	Minutes	Seconds

Leveraging Containers

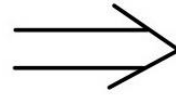
For Highly-Available Applications

You don't have to run any redundant instances of your application!

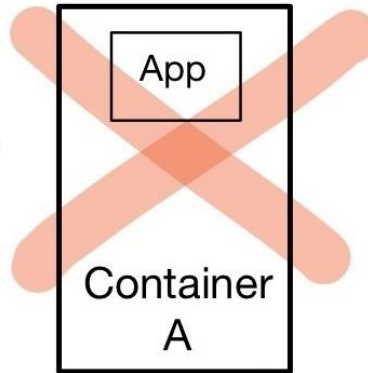
Normal Operation



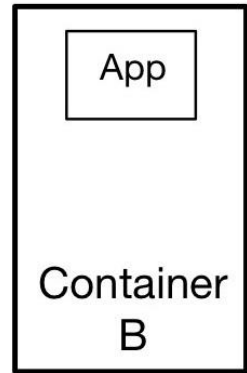
Active



Failover



Failed



Active

Journey to Decarbonization



You don't need **redundant**
servers on the **cloud**!

Modern Cars With Stop/Start



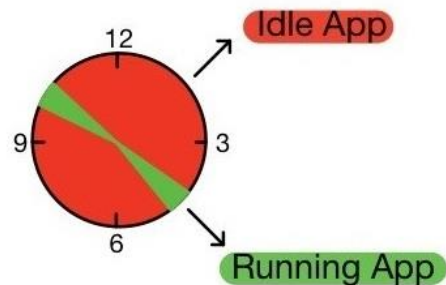
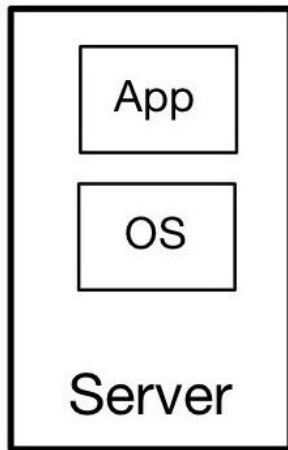
- ▶ Improved **fuel efficiency** by stopping the engine when not needed:
 - Traffic jams
 - Red traffic lights

Consuming Resources

Although the Application is Idle

Is your application waiting most of the time, without doing any actual work?

Long-Lived App



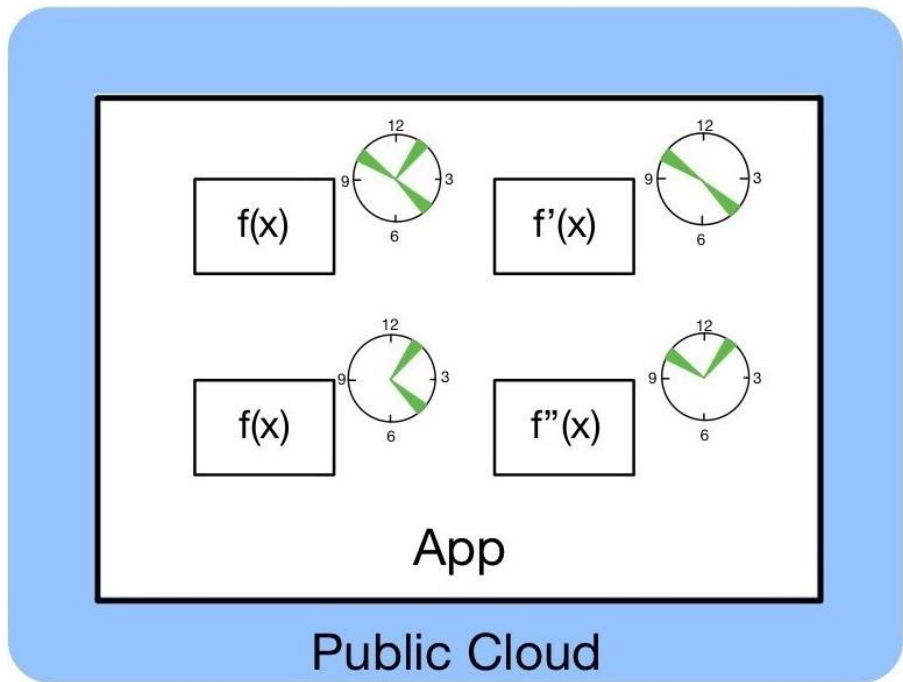
Rearchitecting Applications

That Are Idle Most of the Time

App as Short-Lived Functions

Functions are triggered by:

- HTTP requests
- Events (e.g., file upload)
- New messages in queue
- Tasks scheduled at a given time (cron jobs)



Serverless Functions

Run Your Code Without Managing Servers

- ▶ Automatic **scaling** of individual functions
- ▶ Your code **consumes energy only while processing requests!**
- ▶ Supported languages in Azure:
 - Java
 - C#
 - F#
 - JavaScript
 - TypeScript
 - Python
 - PowerShell



AWS
Lambda



Azure
Functions



Google Cloud
Functions

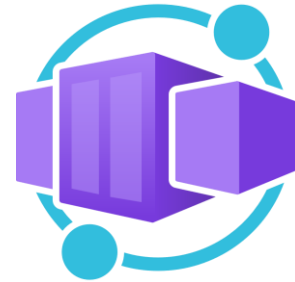
Serverless Applications

Run Containers Without Managing Servers

- ▶ More flexibility than Serverless Functions:
 - Run code written in **any language** (packaged in a container)
 - Support **long-running tasks** (days, weeks)



AWS
Fargate



Azure
Container
Apps



Google Cloud
Run

Journey to Decarbonization



Introduce a **stop/start** mechanism in your software applications

Green Cars

Transitioning to Environmentally-Friendly Cars



► Car **fuels** make the difference:

- Diesel
- Petrol
- LPG
- Electricity

► The “fuel” for software apps → **Programming languages**:

- Compiled
- Running on VMs
- Interpreted

Programming Languages

Language	Type	Energy
C	Compiled	1.00
Rust	»	1.03
C++	»	1.34
Ada	»	1.70
Java	Virtual Machine	1.98
C#	»	3.14
JavaScript	Interpreted	4.45
TypeScript	»	21.50
PHP	»	29.30
Ruby	»	69.91
Python	»	75.88
Perl	»	79.58

Compiled languages tend to be the fastest and **most energy-efficient** ones.

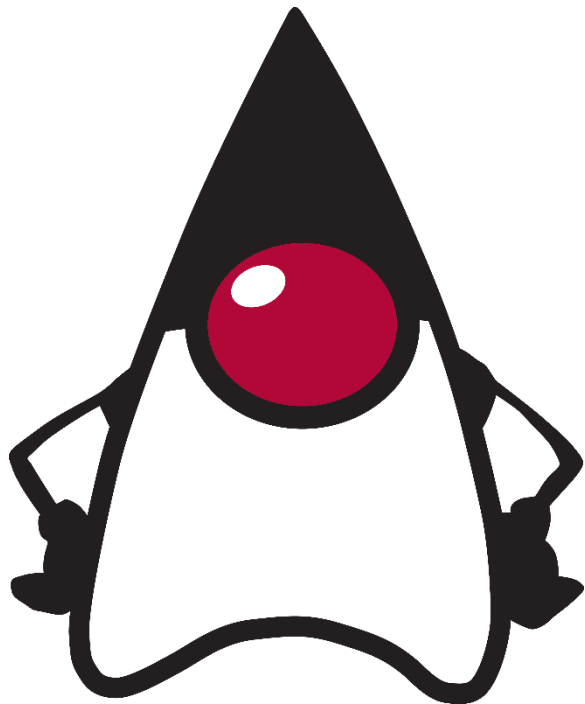
Source: [R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. Fernandes, J. Saraiva, "Energy efficiency across programming languages: How do energy, time, and memory relate?", International Conference on Software Language Engineering, Oct 2017.](#)

Journey to Decarbonization



Identify **energy consuming components**,
used by many users,
and rewrite them in an
energy-efficient language

Java in the Spotlight



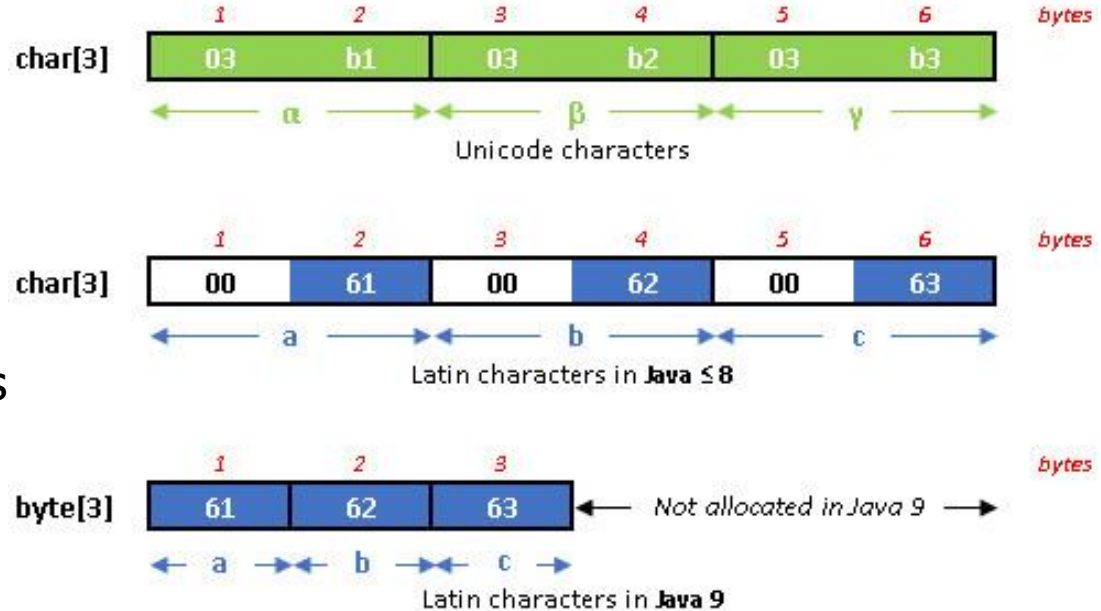
- ▶ Java ranks as the 2nd primary language used by developers
- ▶ Java 8 is used by 72% of them
 - Latest release: Java 18 (March 2022)

Source: JetBrains (2021), State of the Developer Ecosystem Survey
<https://www.jetbrains.com/lp/devecosystem-2021/>

Use the Latest Version of Java

To Benefit From Smaller Memory Footprints

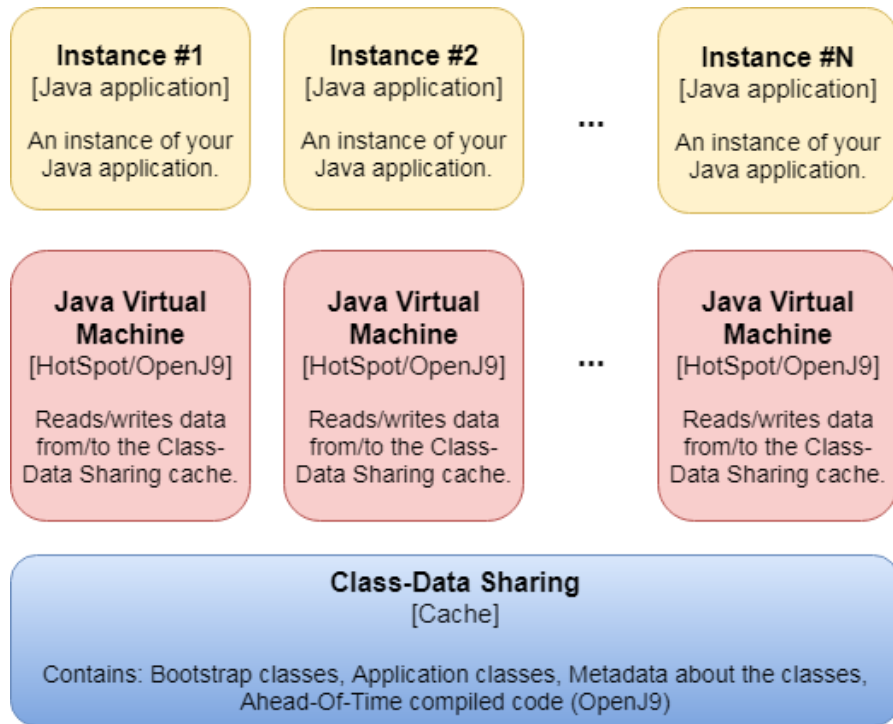
- ▶ Since **Java 9**: enhanced memory allocation for String objects
- ▶ **Memory footprint** & performance improvements of **up to 10%**



More info: <https://www.kolaxis.dev/will-your-applications-run-faster-with-java-9>

Use the Latest Version of Java

To Benefit From Smaller Memory Footprints



- ▶ Since **Java 10**: applications running on the same host can leverage **Class-Data Sharing (CDS)**
- ▶ Reduce memory footprint by **~15%**

More info: <https://www.kolaxis.dev/reduce-the-cloud-bill-of-your-java-applications>

Journey to Decarbonization



Upgrade to the **latest version**
of Java

Less is... Green!

Reduce Your Cloud Bills for Sustainable Software



- ▶ Cloud bills depend on:
 - Network traffic,
 - Used resources (CPU, RAM) x Time
- ▶ Less money spent on cloud bills
 - More energy-efficient software

<Your Cloud> Well-Architected

Frameworks of Best-Practices

- ▶ A **Critical score (0-33%)** in Azure Well-Architected implies an **energy-inefficient** application.



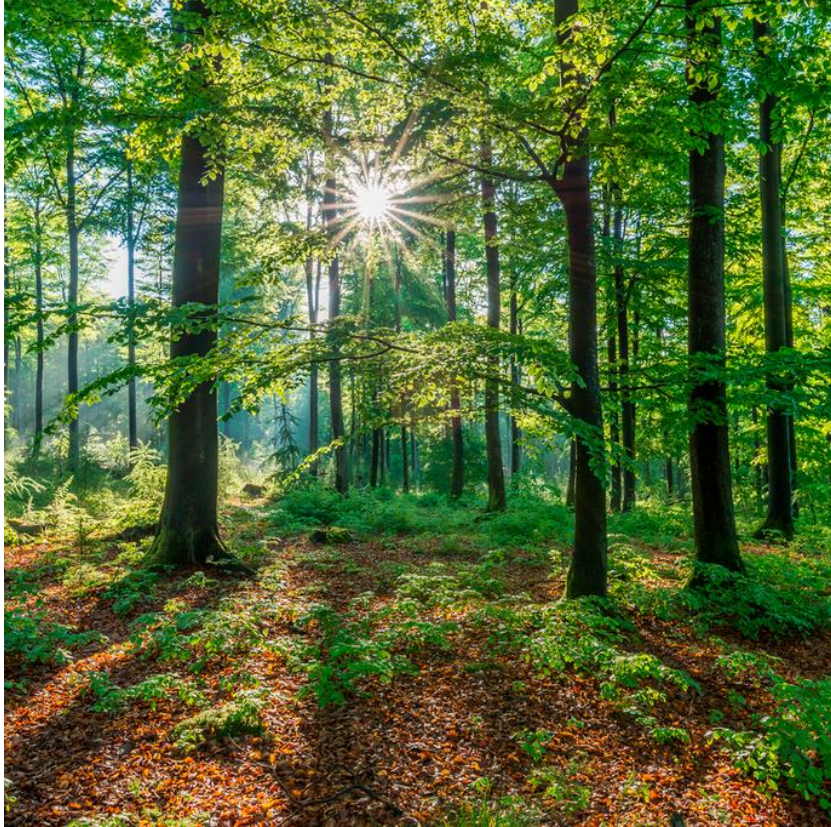
Take the assessment here:

<https://docs.microsoft.com/en-us/assessments/>

AWS, Azure & Google Cloud all have **Architecture Frameworks** that assess your application against:

- Cost
- Performance

Sustainability is the newest “-ility”



- ▶ Include **sustainability** in your discussions:
 - Availability,
 - Scalability,
 - Maintainability,
 - ...

The Green Software Book

Wrote a practical guide for
developers & architects



kolaxis.dev/green



Your Journey to Decarbonization

Reduce **cost** of running
your applications on cloud!

Adopt an energy-efficient **high availability model**
(**Containers** vs **redundant servers**)

Reduce **data** transferred over the **network**

Upgrade to the **latest version of Java**

Rearchitect **long-lived apps** to
short-lived functions

Rewrite components
consuming **most energy**,
using **efficient languages**

What actions will you take to help prevent climate change?



Tweet your answer!

Thank you



@IoannisKolaxis



kolaxis.dev

Atos, the Atos logo, Atos|Syntel are registered trademarks of the Atos group. April 2021.
© 2021 Atos. Confidential information owned by Atos, to be used by the recipient only.
This document, or any part of it, may not be reproduced, copied, circulated and/or
distributed nor quoted without prior written approval from Atos.

Atos