



AutomaticTM

Let's Automate Business.



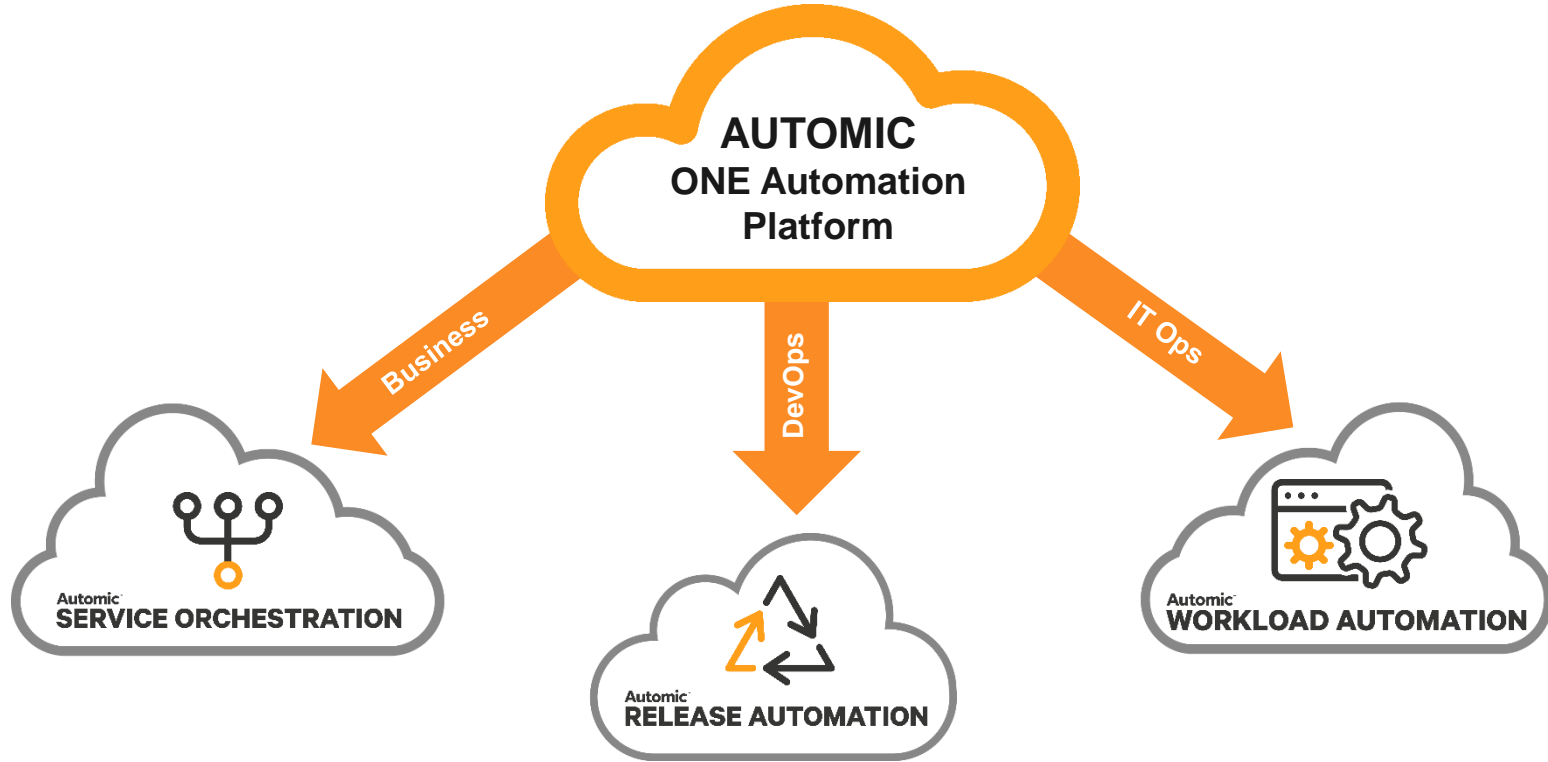
November 12th, 2016

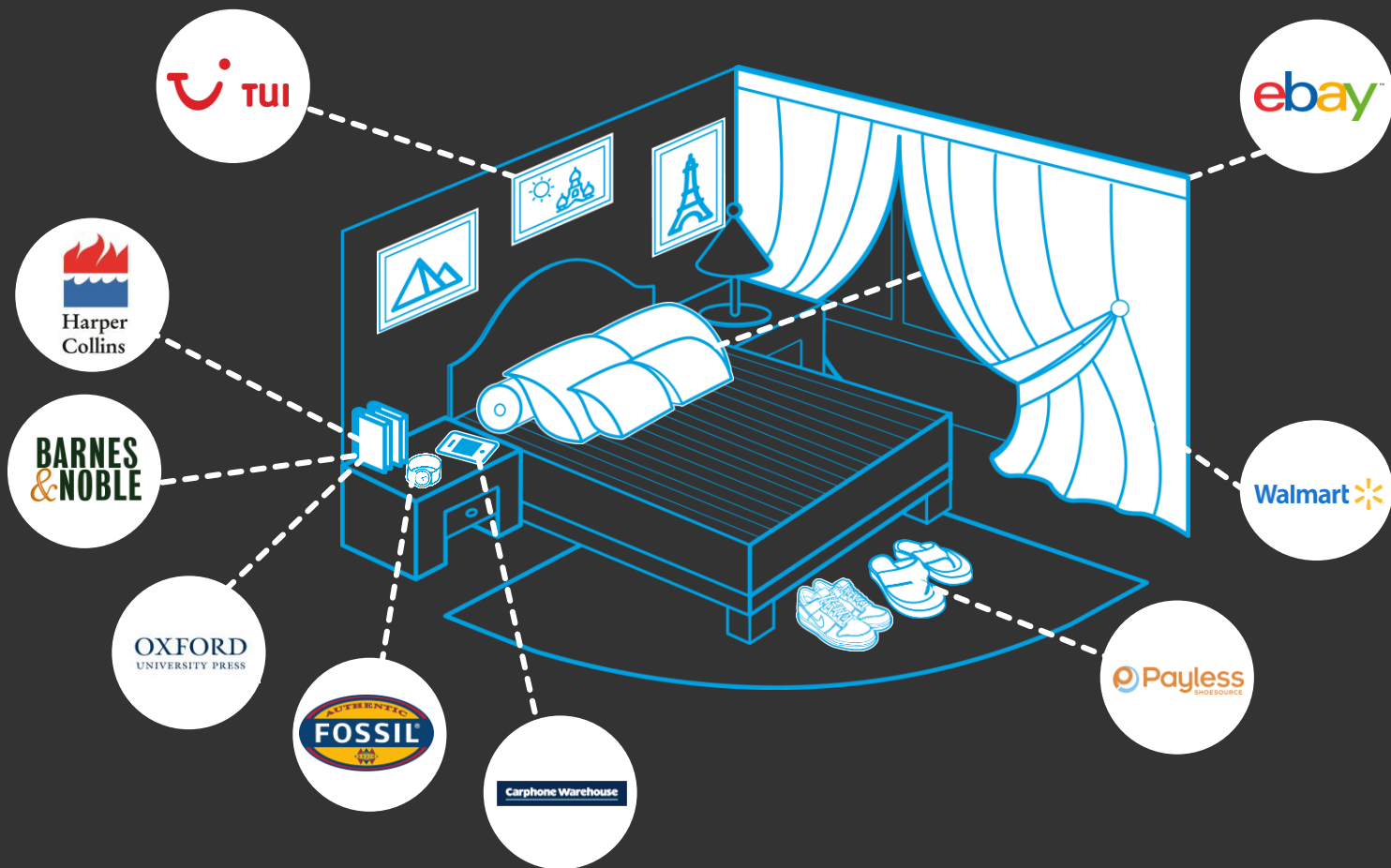
Continuous Delivery in Complex and Heterogeneous Environments

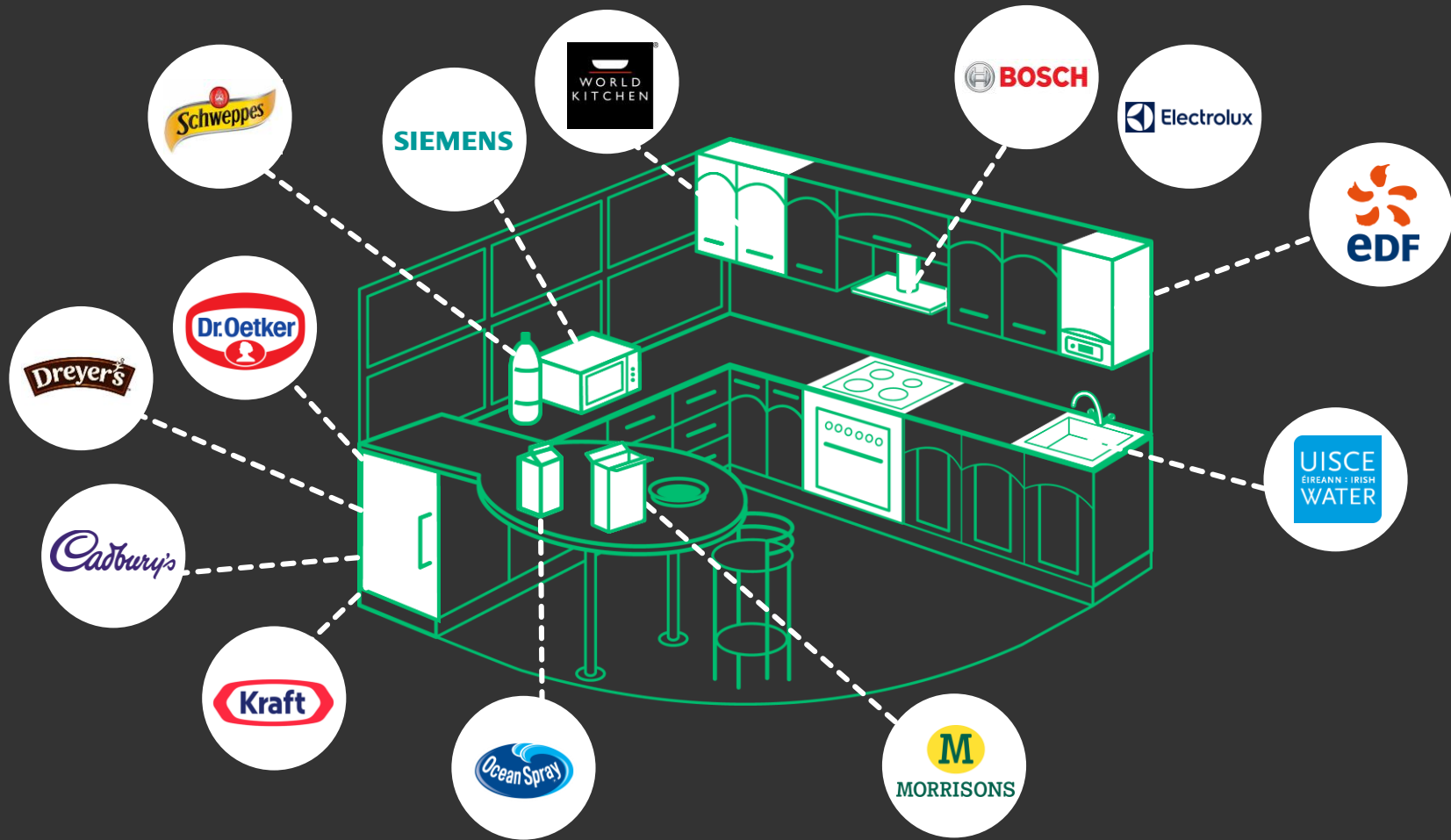
Josef Puchinger, VP Engineering

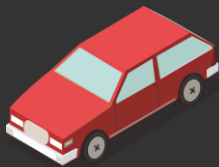
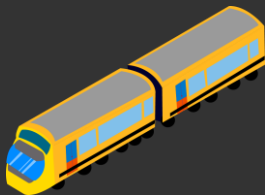
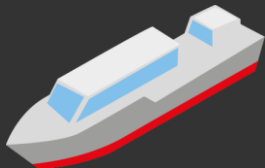
Christopher Hejl, Chief Application Architect

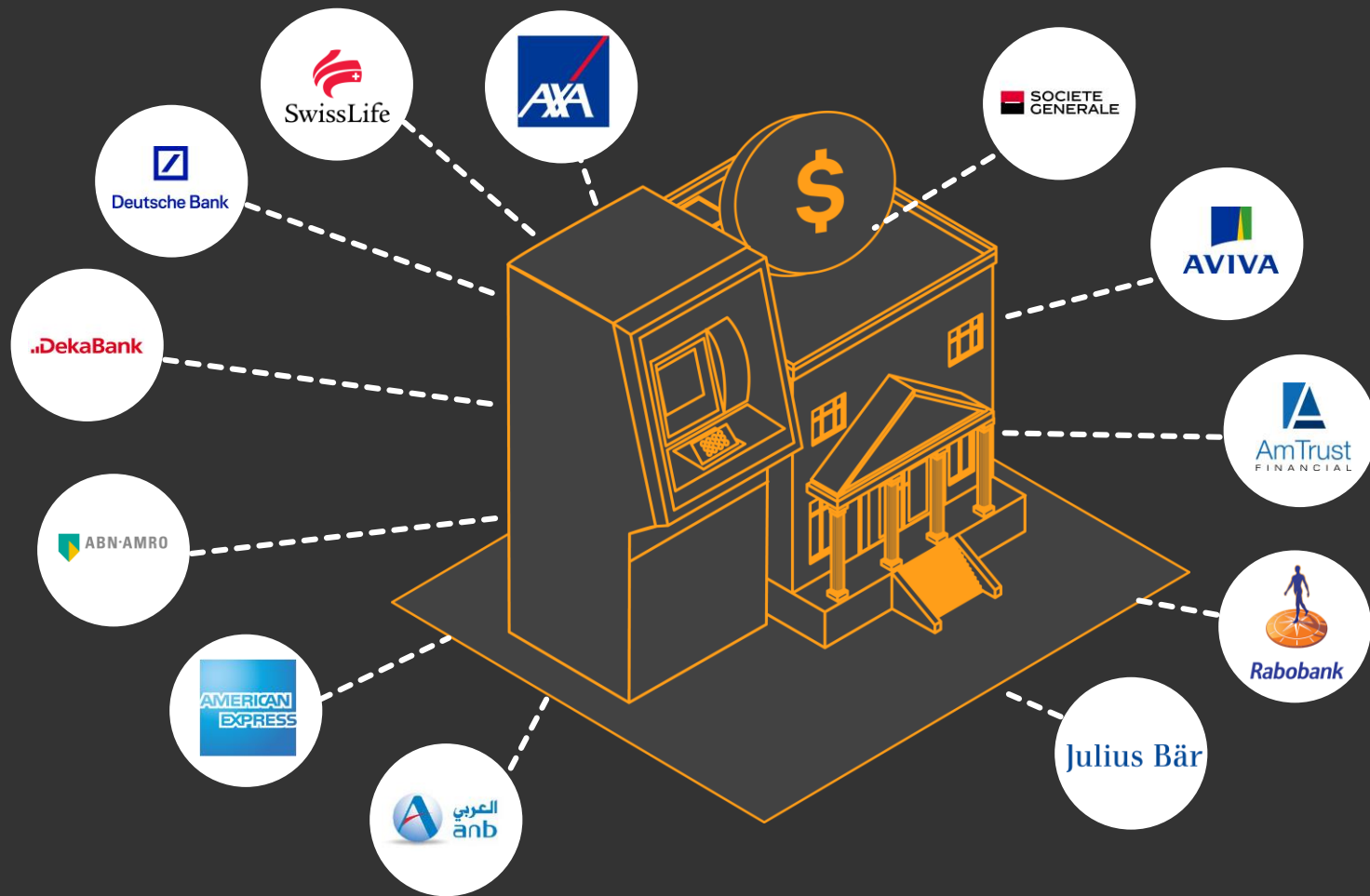
Atomic in 2 minutes

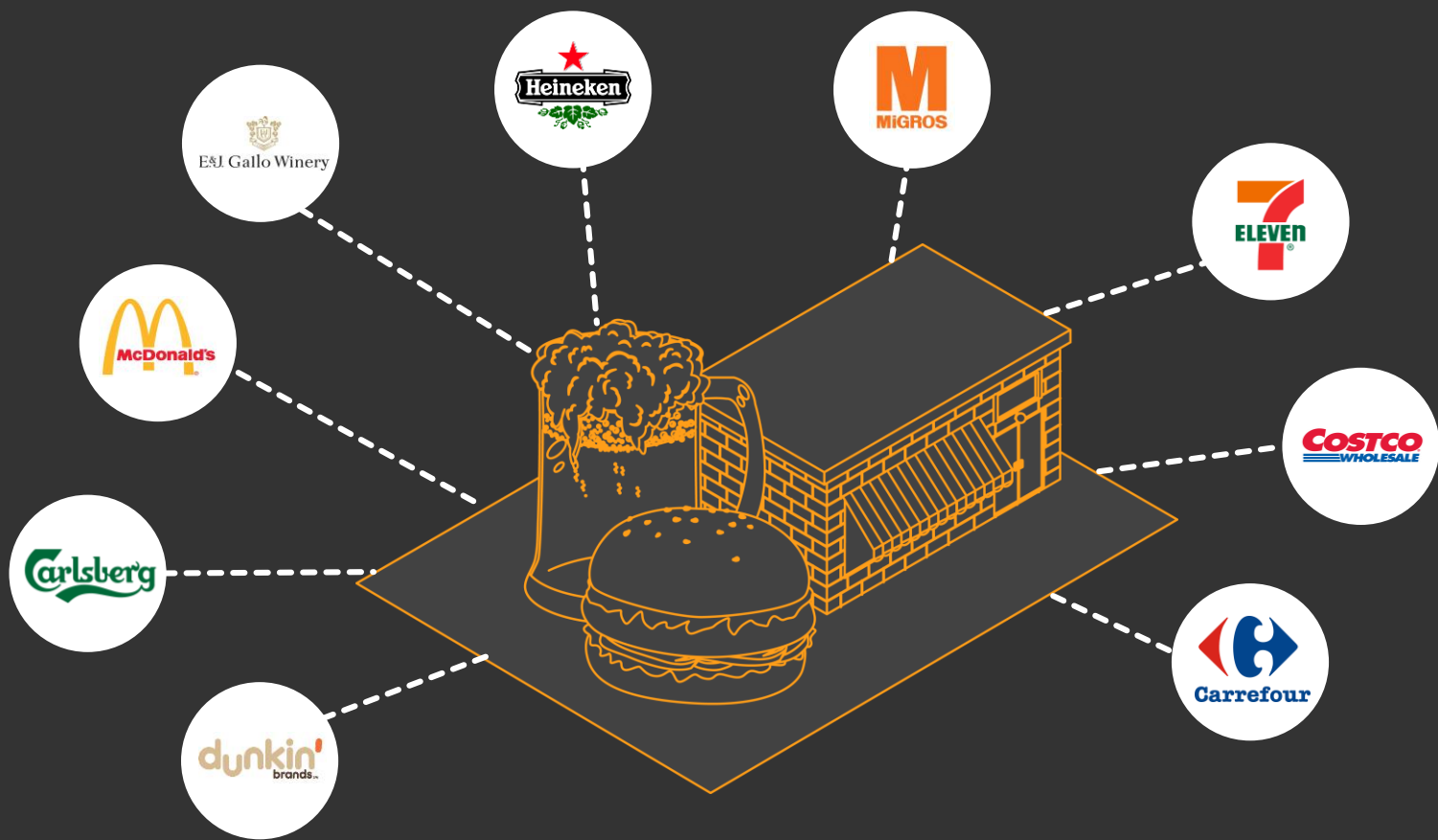




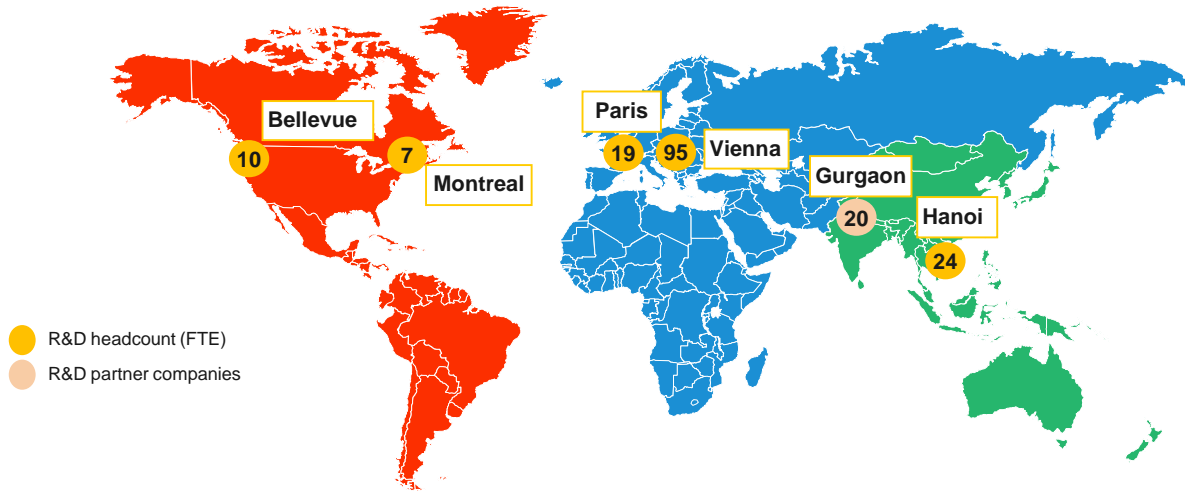








Overview of the R&D Organization



R&D principles:

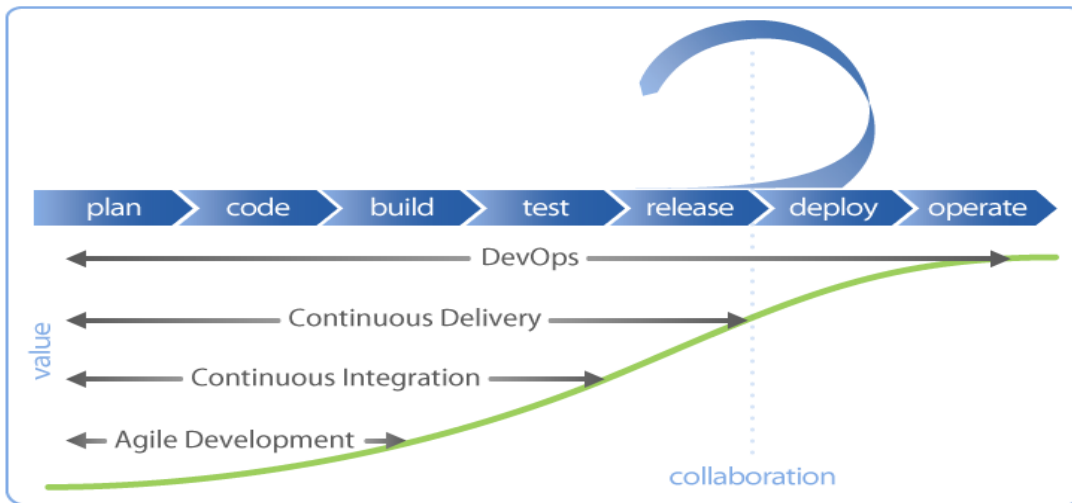
- x-functional teams
- Feature teams
- SCRUM and KANBAN
- Automation

The 3 pillars to become faster

1. Agile Software *Development*

2. Agile Software *Delivery*

3. Reduced Software *Complexity*



Why Continuous Delivery?

Quality

One way to **help developers focus on creating more stable systems** might be to measure the effect of releases on the stability of the affected systems.

(source: unknown)



Risk Mgmt.

Is your style of delivery high-risk, 'big bang' deployment? Unless you're an adrenaline junkie, **you're just risking spectacular failure** with your company's money and your sanity.

(source: Jez Humble, co-author of Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation)



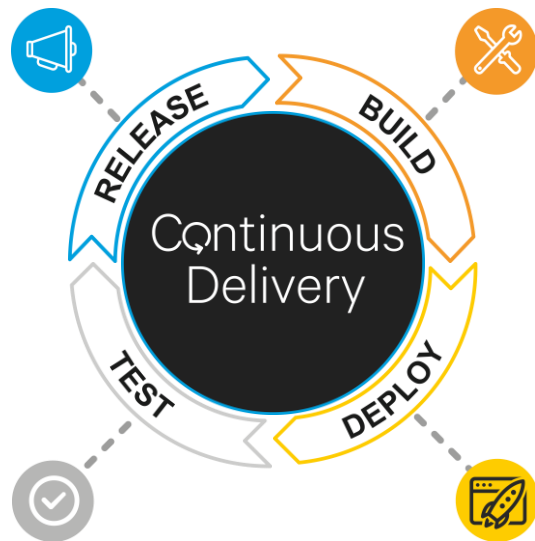
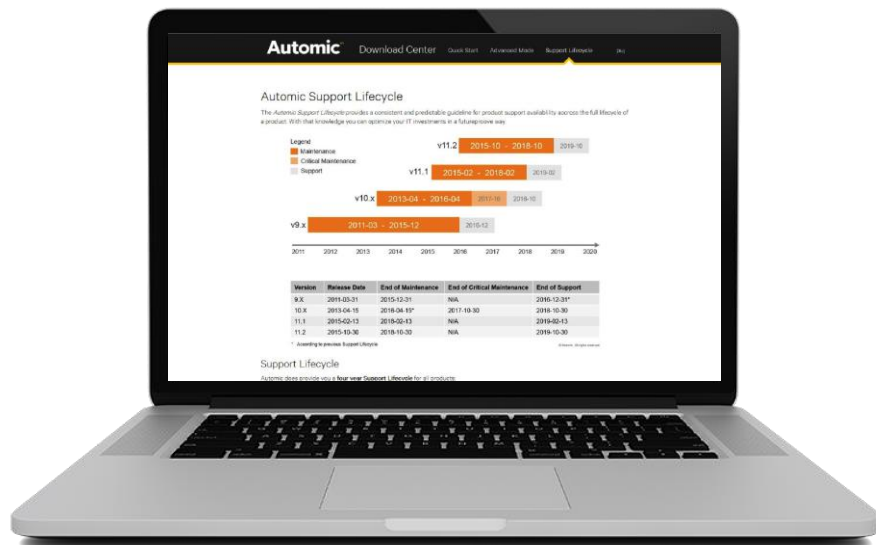
Time to Value

Make your Potentially Shippable Increment **available for internal production use, demos** and POCs.

(source: Josef)



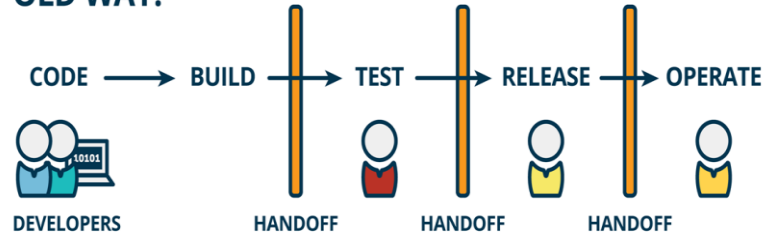
Continuous Delivery vs. Continuous Deployment



Change where and when needed ...

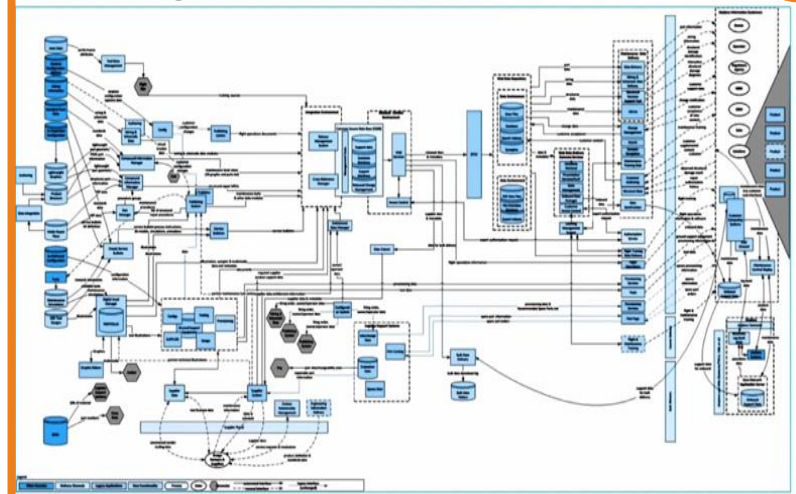
First

OLD WAY:



Step
by
Step

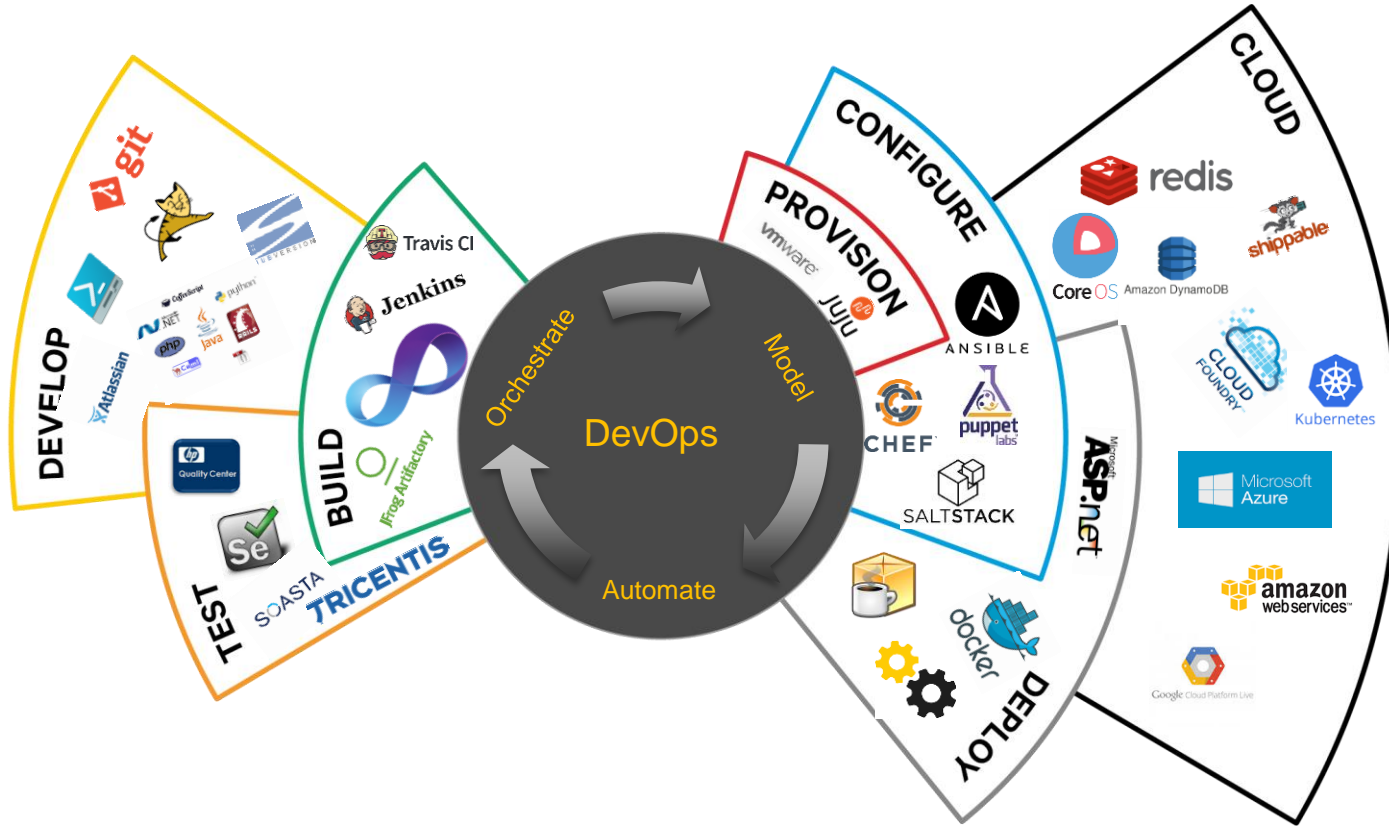
Real enterprise IT looks like this



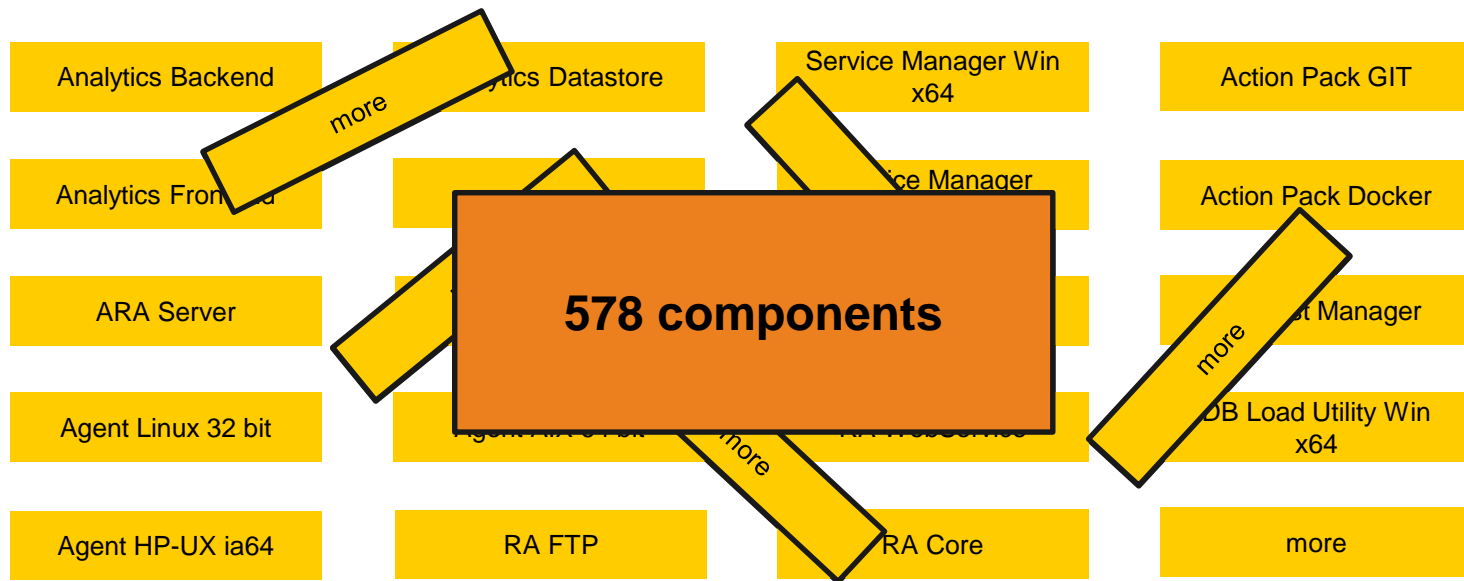
First



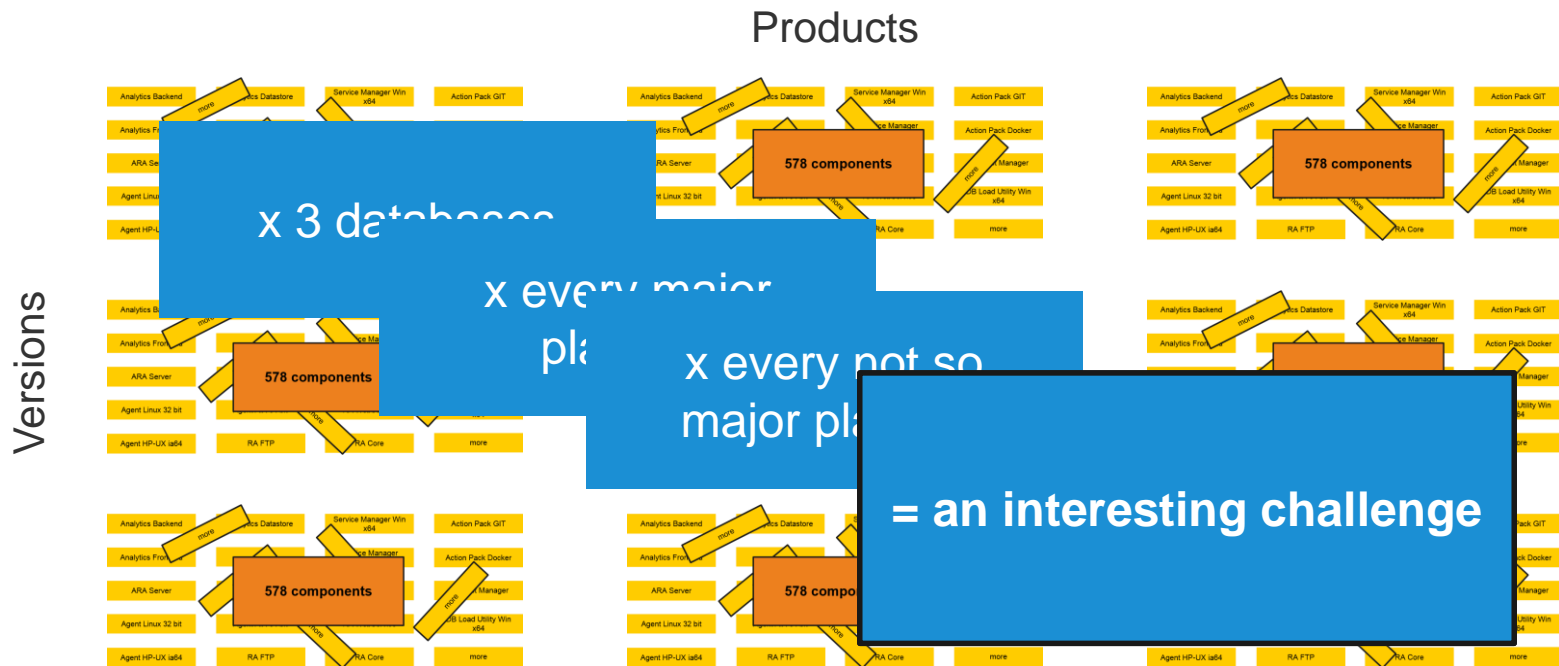
... but - Developers - keep your tools!



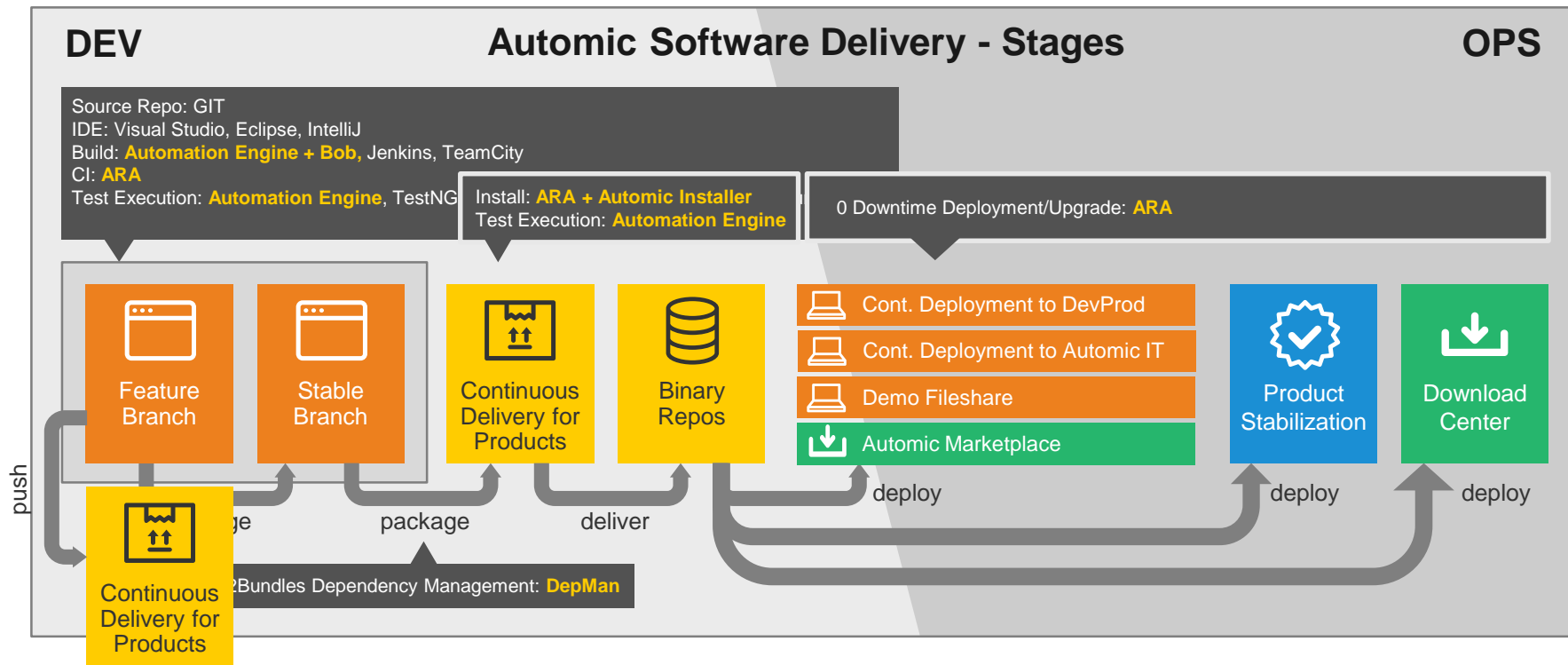
Components and Products



Components and Products



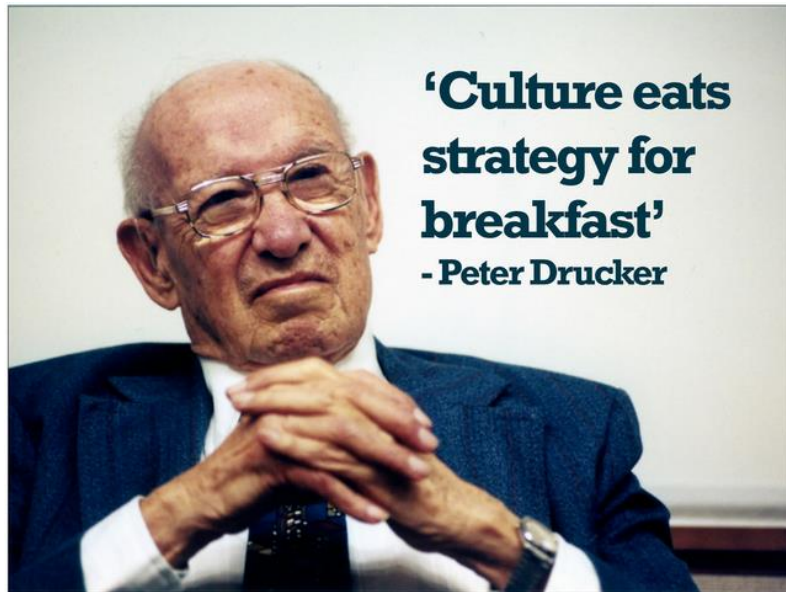
How does it work in detail?



The challenges

1/78

- Culture & mindset: merging to product master codeline is *the developer's go live*
- Keep the feedback loop *short*
- *Effective* test automation
- Full stack *application provisioning* for heterogenous and complex applications
- Make people *beg* for the new software increments
- I can't do this because it does not work for ...



Our benefit is our customer's benefit



Better features and usability



Better software quality - early detection of integration & compatibility issues
Big customers are involved and upgrade earlier



Better install & upgrade procedure



Releasing software *is becoming*
a pure business decision



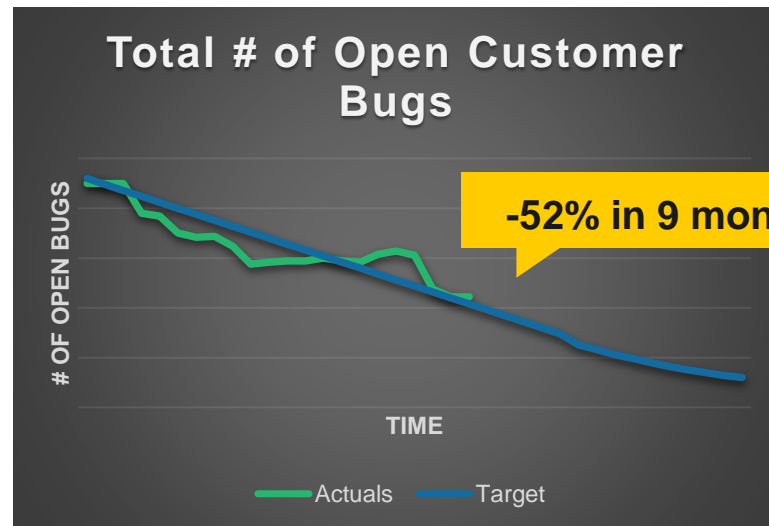
Quality



Risk mgmt.



Time to Value



Our Products are **great** because

1. we do **Continuous Delivery.**
2. we do **Continuous Deployment.**
3. we do it with **Our Products.**



Continuous Delivery
Can you *survive* without it?



Josef Puchinger
josef.puchinger@automic.com



AutomicTM
Let's Automate Business.



Christopher Hejl
christopher.hejl@automic.com