

Industry

Applicable to all traditional architectures such as "request/response" pattern simply cannot meet the challenges of real time and extreme scale.

Website

Fastcube.fr

Author

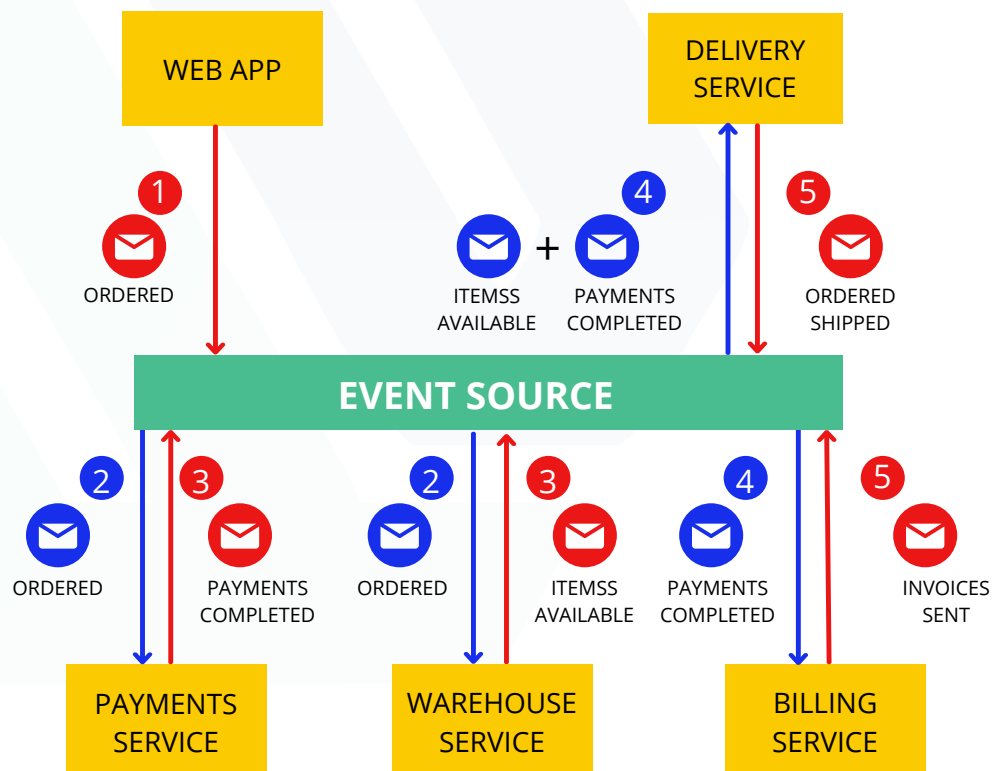


Michel Thai,
Data Architect Lead

INTRODUCTION

The world is changing. Companies now run global businesses that span the globe and hop between clouds, breaking down silos to create seamless applications that work together for the good of the organization. This continuous state of change means that legacy architectures are insufficient or unsuitable to meet the needs of the modern organization. Applications must be able to run 24x7 with 5-9s (uptime of 99.999%), as well as be superelastic, global and cloud native.

Traditional architectures such as "request/response" pattern simply cannot meet the challenges of real time and extreme scale. Event-driven architecture breaks everything down into separate microservices that create constant streams of data events that can be received and processed much faster. Event driven architecture isn't just for streaming data. It brings added efficiency, flexibility, and security to your systems, making them more maintainable, scalable, and robust.



WHY EVENT ?

- It is Simple:

Event is not a new concept and it happened all the time and is still happening, they are everywhere.

But only what's being observed are modelized and we are observing the change of states.

Event model are abstract and is up to the observer to defined what detail are needed/useful.

- Found in everyday life:

There are events even with coffee machine: From state OFF to READY => event « initialization completed» from state READY to BREWING => event «brew button/command triggered ».

From state BREWING to READY => event « brewing process finished » (!!! Finished does not mean everything went ok you have to check its results to know that).

Workflow is commonly used in Business to describe an choreographed and repeatable pattern of activity in an Organization inside which event can be found.

- Experience are built from past events:

Event are fact what's already happened; it is immutable thus can be used as log for monitoring historical data can be used to identify weakness in system and provide opportunity to enforce resiliency also known as continual improvement process process efficiency can be improved with design review etc....

WHAT CAN BE ACHIEVE WITH?

- Shared knowledge:

Event could be used to ease the understanding between IT and Business.

- Build Resilient System:

Monitoring of Events can be used to provide self-healing features

- Implementation and reuse of existing code :

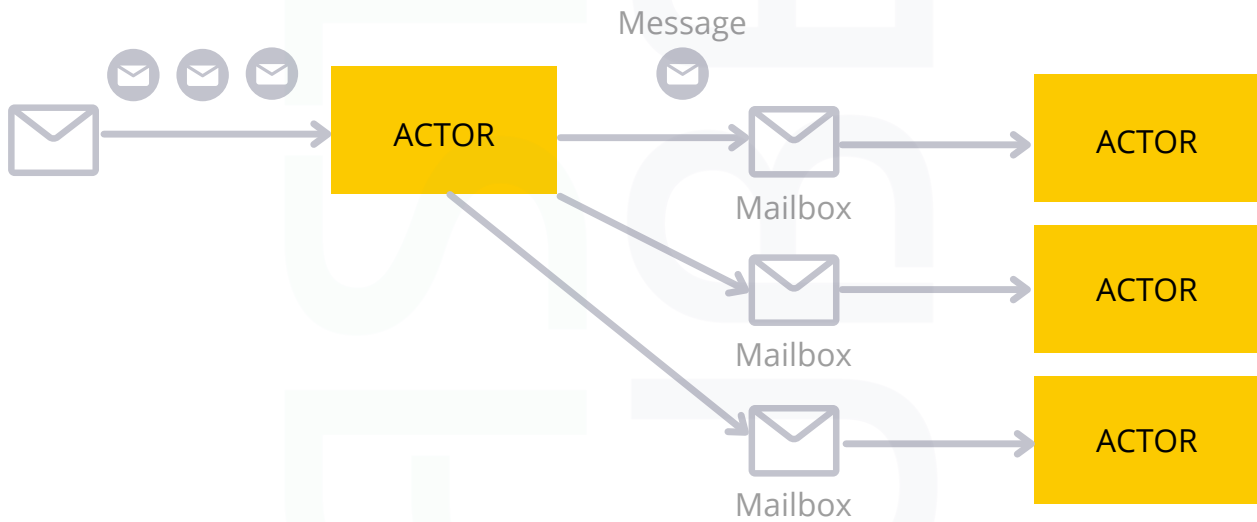
- Many actors can consume the same events.

- Update of component can be done with minimal interruption of services.

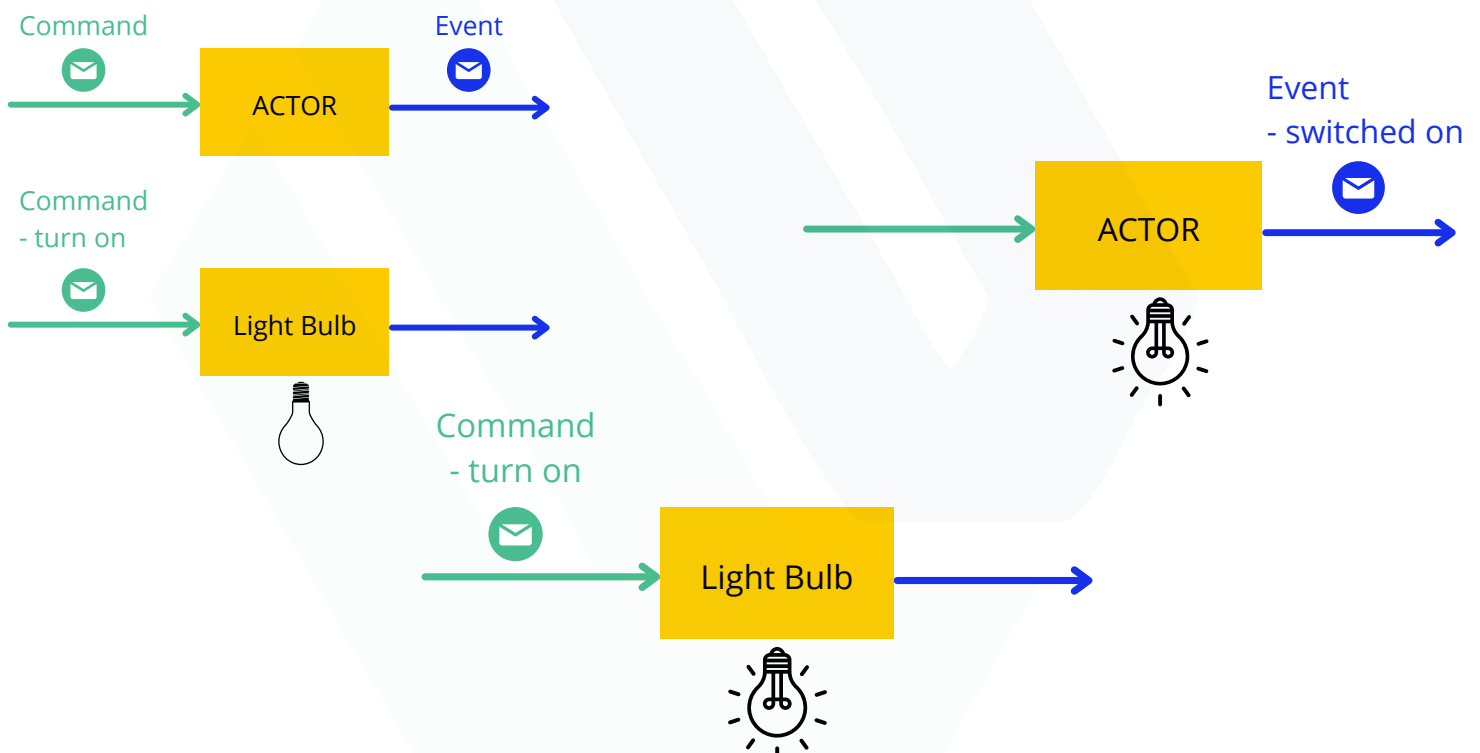
- Events allow design of independent modules which in turn speed up productivity.

HOW TO BUILD SUCH SYSTEM?

Actor model as foundation



- The **ACTOR MODEL** in computer science is a conceptual model use to deal with concurrent computation.
- **ACTOR** is an abstraction that represent a unit of calculation and consume messages
- **ACTOR** can produce Message to others actors.
- **ACTORS** never have direct interaction between them but indirectly through message posted in a MailBox represented by their address.
- **Actor** consume **1 Command** at a time.
- The process of a **Command** can generate an **Event** every time the observation of a state has a change.
- One **Command** can produce multiple **Events**.



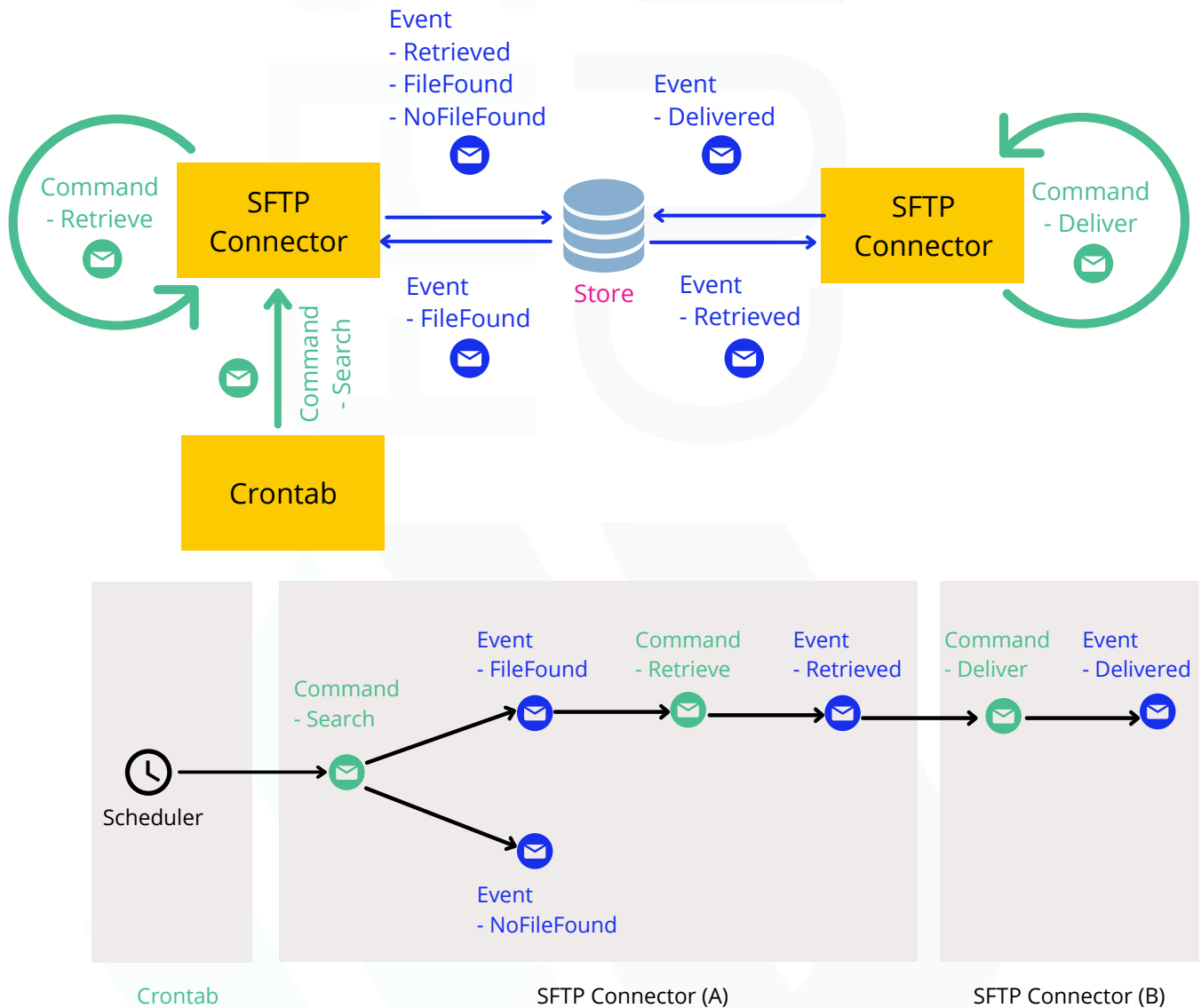
Reactive programming

Reactive programming is the general paradigm behind easily propagating changes in a data stream through the execution of a program.

When x changes or updates in one location, the things that depend on the value of x are recalculated and updated in various other locations in a non-blocking fashion, without having to tie up threads sitting around just waiting for events to happen.

Chained reaction in distributed system

Using Event as a medium of communication between Actors a chained reaction can be defined. This chain can be the representation of the processing of a « business flow ».



What's next ?

In this article, we have seen a short introduction about Event-Driven Architecture, this architecture is an important paradigm shift in a system, it can be designed in multiple approach with combining multiple patterns based on event.