The Akka Actor System

concurrent programming, but cool

XMas Workshop @ SDC:WOB

Felix Wunderlich

„Everything is an actor!“

- actor = computational entity which receives messages and reacts to them by
  - sending further messages to other actors,
  - creating new actors,
  - performing local changes to adjust own behavior.

- result: message-driven model which supports concurrency

The Akka Framework

- implementation of the actor model on the JVM
  - available for Java, Scala, and Kotlin (with adjustments)
  - part of the Play framework
  - license: Apache 2.0
- features:
  - fit for distributed systems
  - transparent remote communication
  - adjustment to use of resources (cpu, memory)

Important commands

- Creating an actor:

  spawn(Santa(), context)

- Sending a message:

  Santa ! OrderPresentsMessage(10)

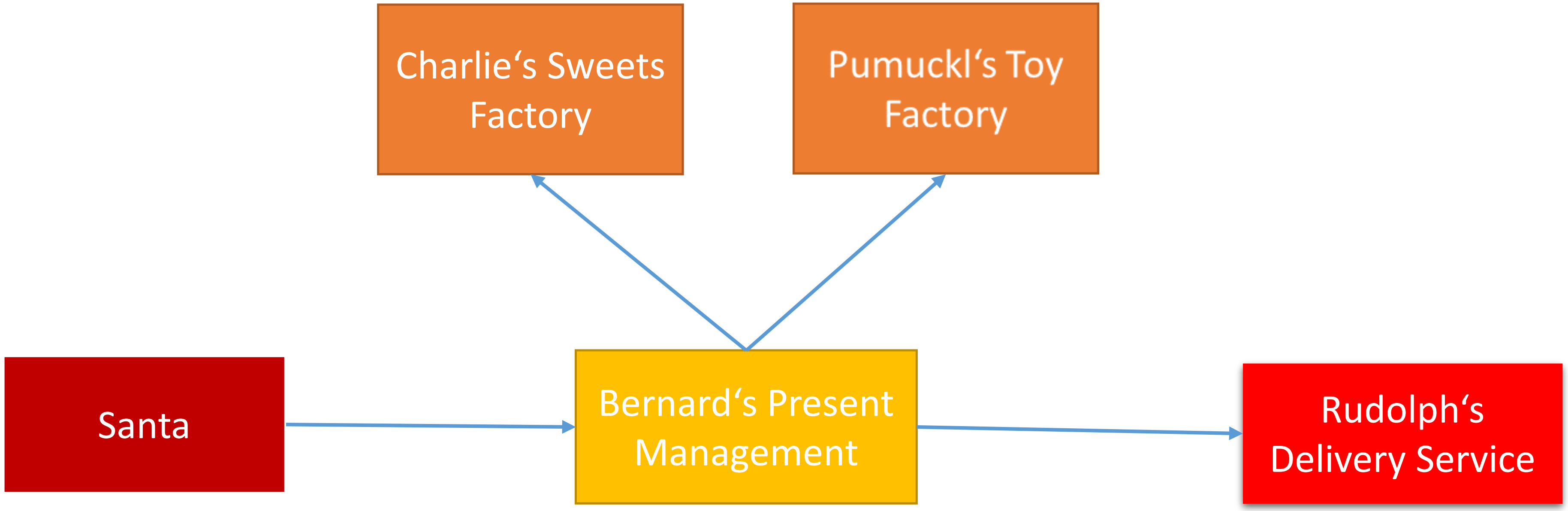
- Sending a message and waiting for a reply:

  val future =Santa ? OrderPresentsMessage(10)

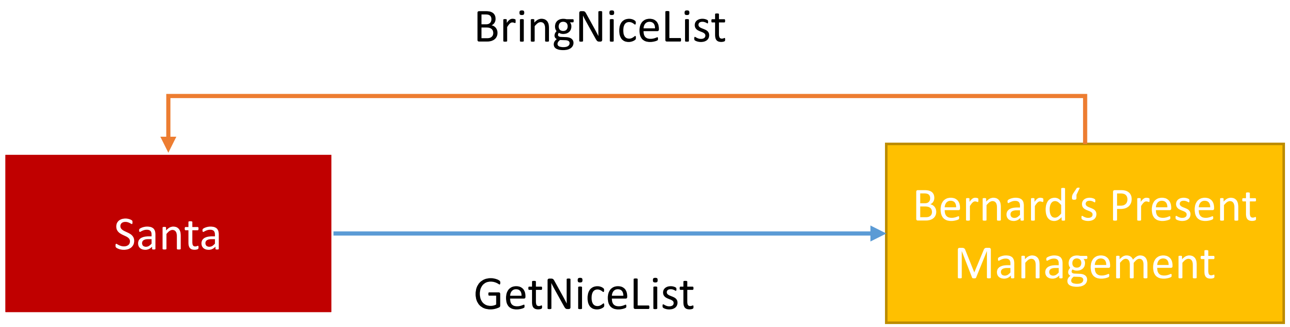
Testing concepts

- Actor state:
    - Goal: on inbound message, check if correct message is sent to another actor
    - Example: *SantaSpec*
- Inter-Actor Communication:
    - Goal: check if correct response is received when sending a message to another actor
    - Example: BernardsPresentManagementSpec

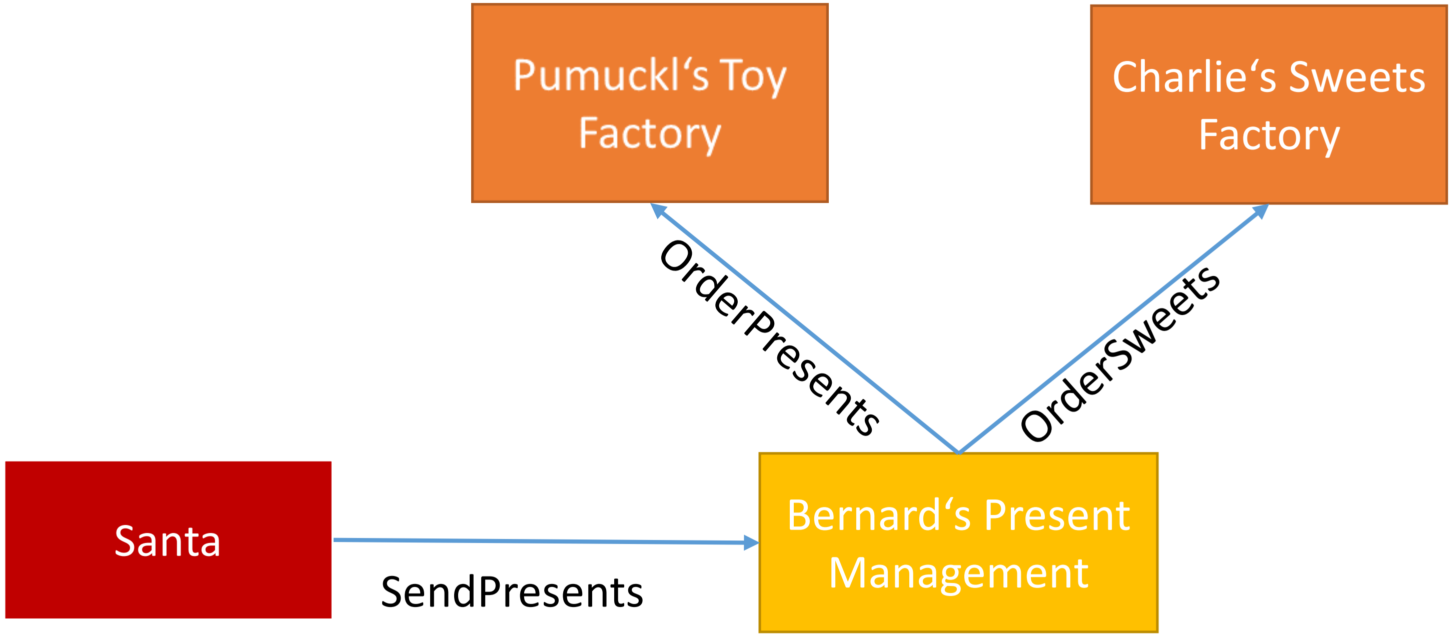
Example: Santa's Christmas Shop



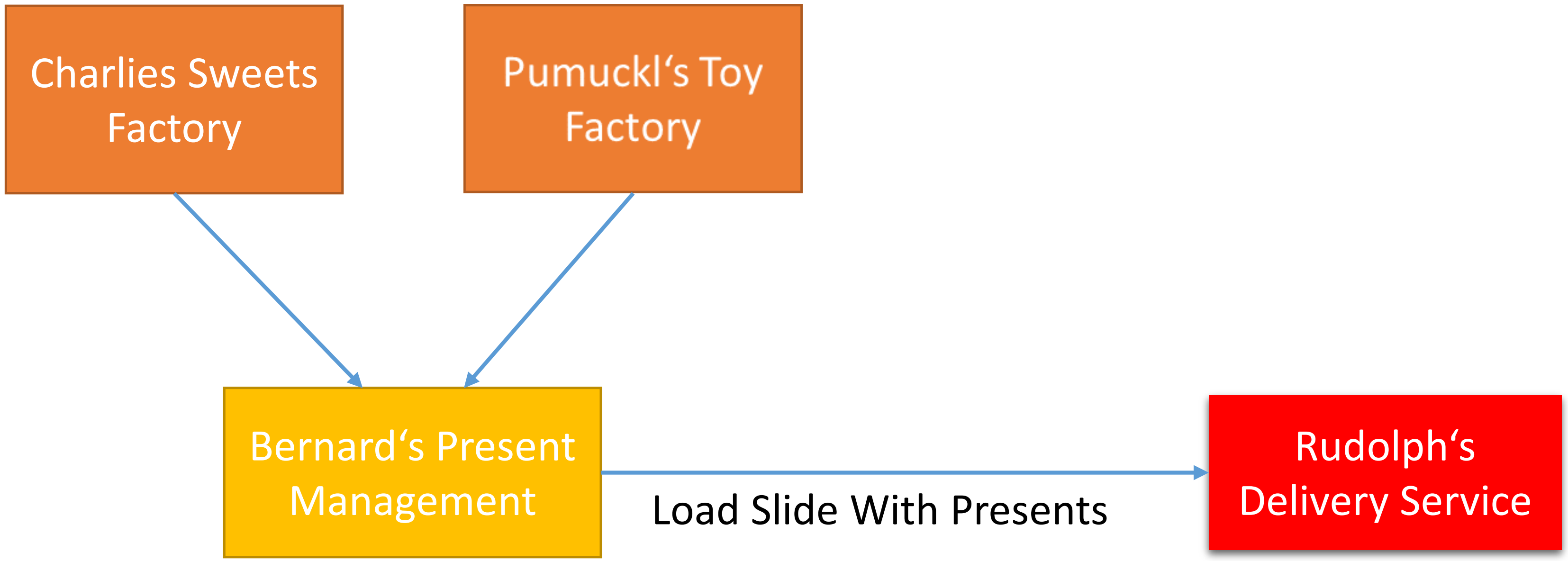
Step 1: Get Nice-List From Bernard's



Step 2: Order Presents and Sweets



Step 3: Load the slide



Charlies Sweets Factory

Pumuckl‘s Toy Factory

Bernard‘s Present Management

Load Slide With Presents

Rudolph‘s Delivery Service