



Think in Erlang!

Viktória Fördős

Erlanger @ Cisco NSO Core

16 May 2019, Code BEAM STO

Which programming language was the first you learnt?

Basic
Java
Pascal
C



Which programming language was the first you used professionally?

Java
Python
C/C++
PHP

Cobol



Which programming language do you work with?

Elixir
Erlang
Other



Imperative paradigm

Imperative

Statements, describing how to execute the program.

Goal is achieved through state changes.

Focus is on the control flow.



Is Erlang an imperative programming language?



Declarative paradigm

What we learn first

- High level
- Quick prototyping & testing
- No mutable objects



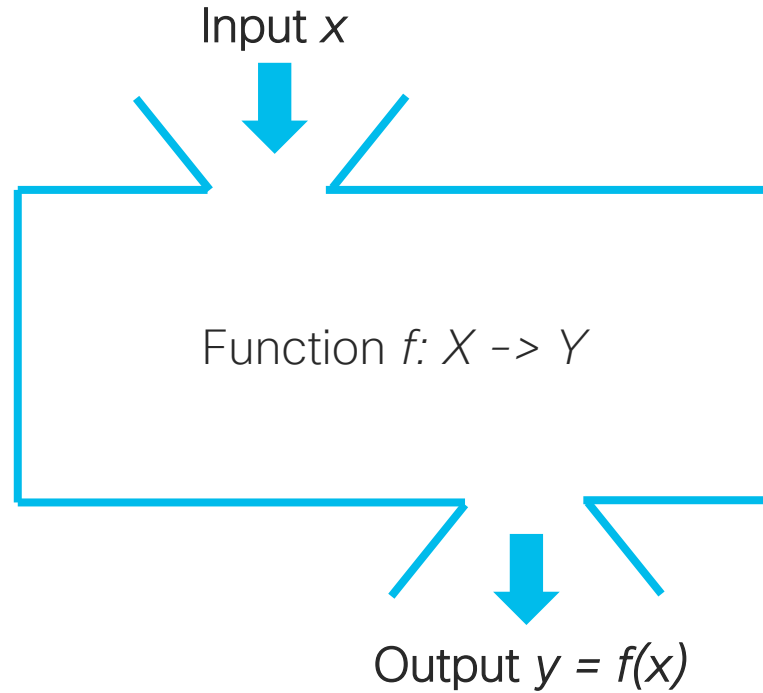
Next, we learn

- Lambda calculus
 - Function application
 - Function abstraction
 - Function composition
 - Higher order functions
 - Fix point theorem



A word cloud consisting of the word "Data" repeated multiple times in various colors (green, blue, red, orange, yellow) and sizes. The largest "Data" is green and centered at the top. Other instances are scattered around it, with some being smaller and some being medium-sized. The colors used include shades of green, blue, red, orange, and yellow.

IO model



Data Transformation Function

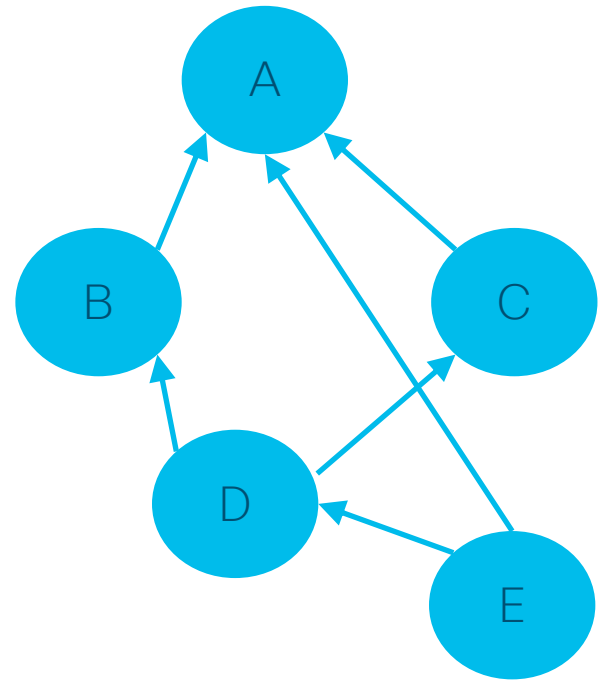
$i(A, h(f(A), g(A)))$

$B = f(A),$

$C = g(A),$

$D = h(B, C),$

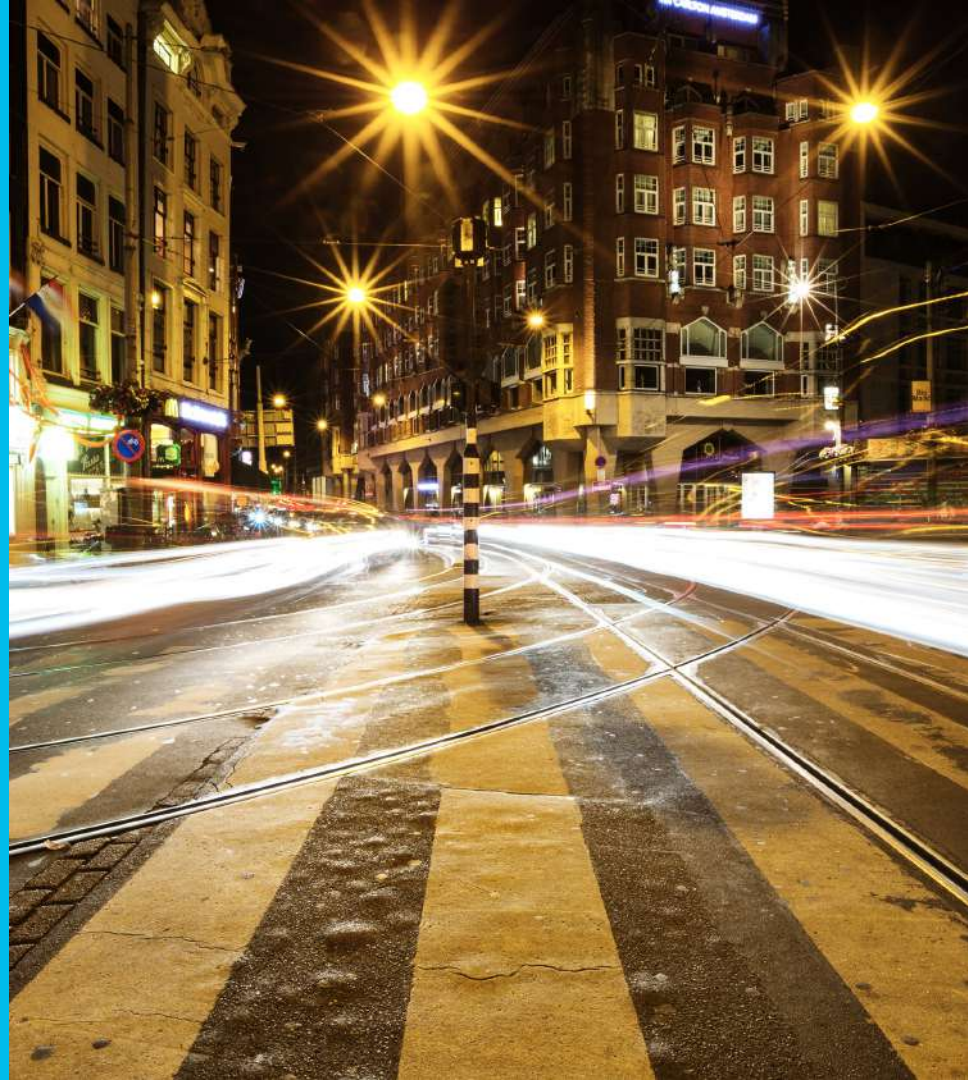
$E = i(A, D)$



Erlang

Concurrency

Based on data dependencies



Process

Evaluates a function.

Has its own world.

Uses signals to communicate
with the outside world.

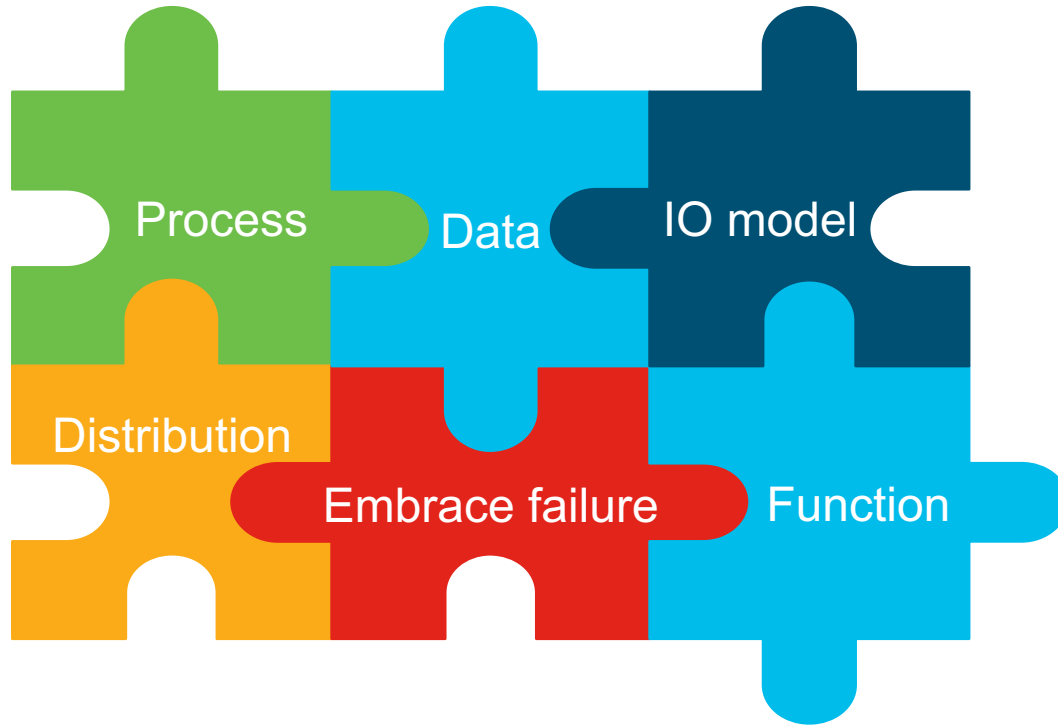


Distributed

Network is not always reliable.
Things will fail. “Let it crash!”



Think in Erlang



Case Study

Intent-Based
Tools



DevOps CI/CD
Pipeline



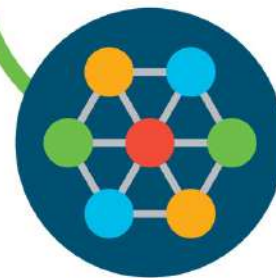
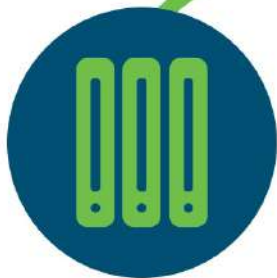
OSS/BSS
Systems



Scripts and
Applications



Service
Orchestration



Physical and Virtual Infrastructure

Network Services

VPNs, ACLs.



The problem

Service lifecycle management.

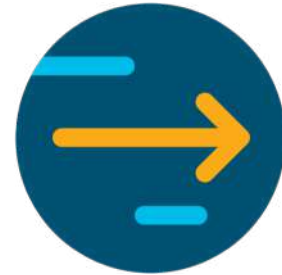


NSO way

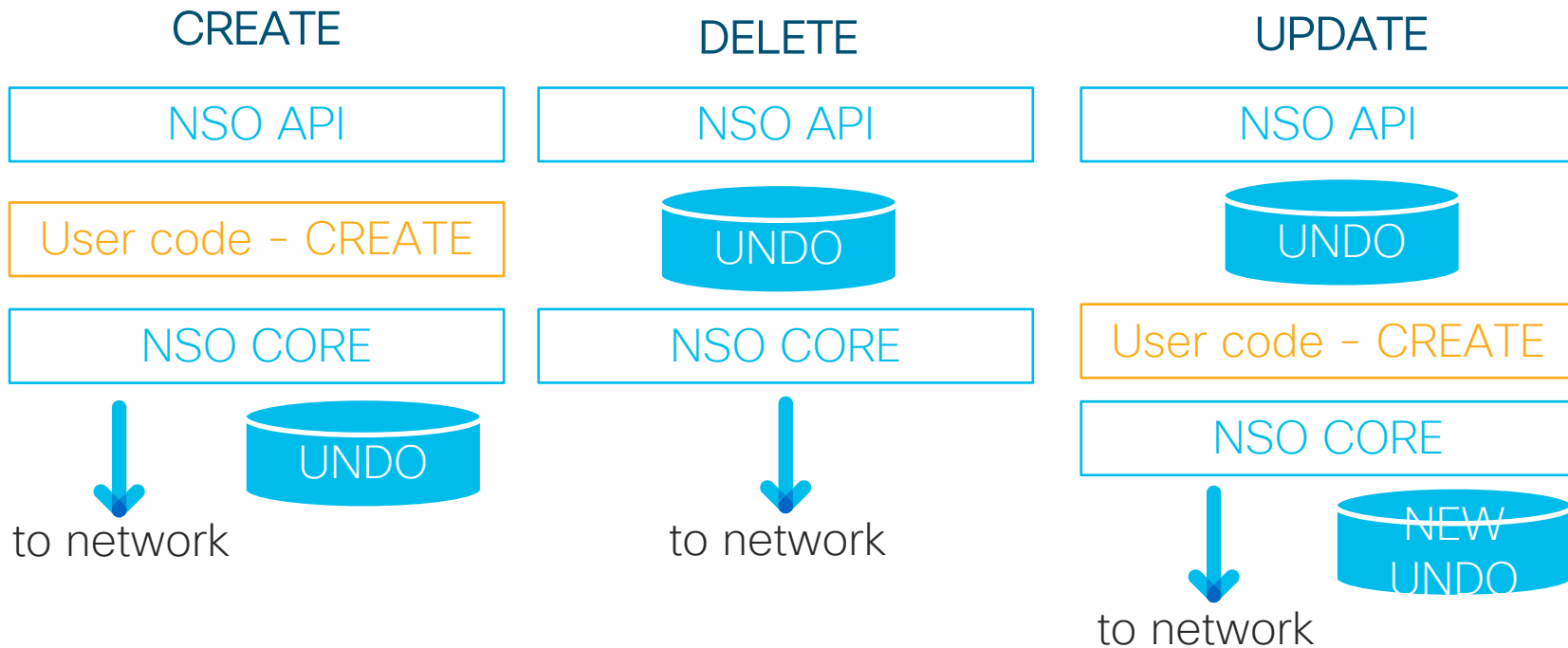
NSO needs

- Service model – IO model
- Mapping from service input parameters to actual device configuration. – Data transformation function.

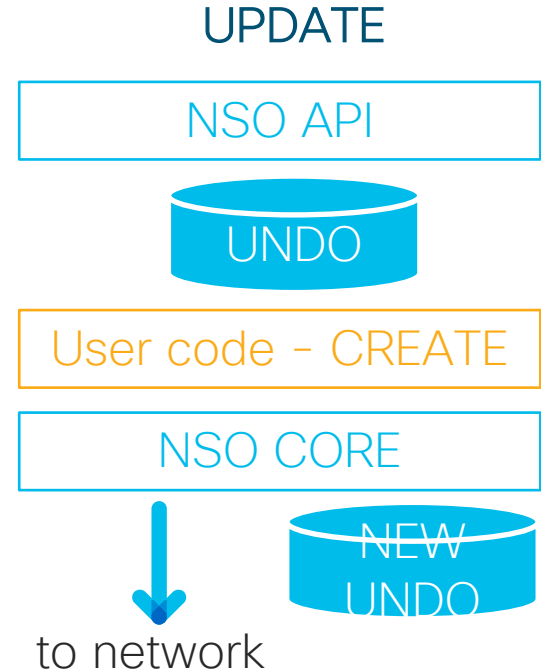
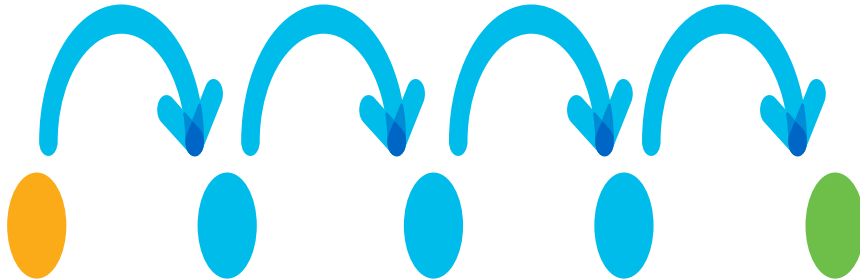
Service lifecycle management:
create, update, delete.



Tail-f FASTMAP algorithm



Tail-f Reactive FASTMAP algorithm



Think in Erlang

