

# BISON

# Bionic Software Networks

Ralf Ulrich

CORSIKA Workshop 2023 @ KIT

# Questionnaire

When you reflect over the role of software in a project, what do you (typically) consider more relevant / interesting:

- Code
- Data
- This is the wrong question

# Types of software

„compiled“	„interpreted“
programs	scripts
coding, compiling, building, running	scripting
runtime: fast	runtime: slow
development: slow	development: fast
fix, once compiled	flexible

# or: Homoiconic software

Definition:

**There is no difference between code symbols and data.**

For example:

- LISP
- julia
- Wolfram Alpha

# Very basic example, LISP

```
(print (eval (+ 1 1)))
```

→ 2

```
(print (quote (+ 1 1)))
```

→ (+ 1 1)

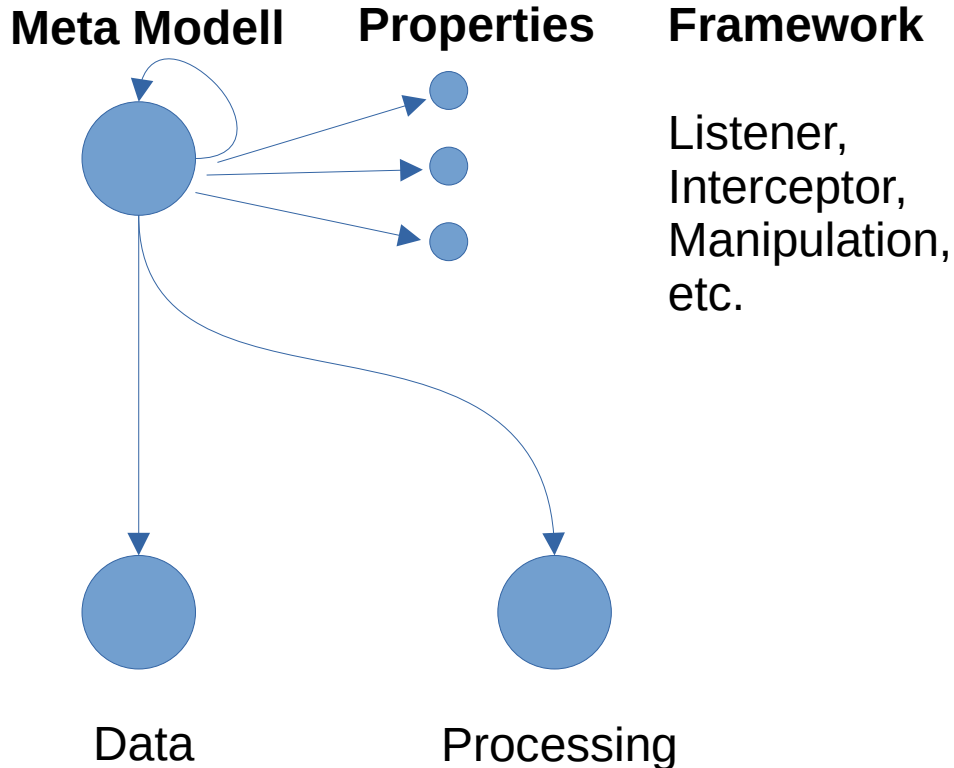
- Symbols are interpreted as data or code depending on context.
- Programs are just a list. Position in list is meaningful.

# → Our goal

- Maximum focus on the data-quality of code
- Minimize „syntax“
- Focus on relation of things between each other
- Full type system, auto-completion, expressive:

→ Modelling

# „Generic Modelling“



- Reflection of everything
- Dynamic creation and modification of types
- Just-in-time compilation
- Full run-time control

# Very basic example

## model Person

- property Name : String
  - property Friend : Person
  - property Age : int
- Properties can be browsed and manipulated at runtime.
  - Interceptors can be installed on any **model**, **property**, or concrete **entity**.



# Code as Data

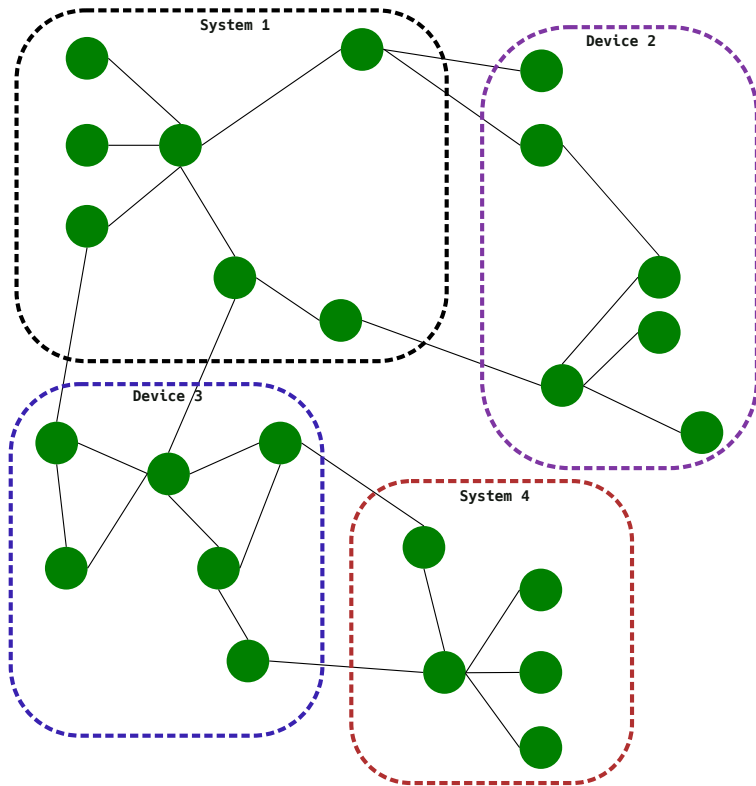
- e.g., use the same „mechanism“ to transport code to data, or data to code (e.g. „grid“, but also: mobile, IoT, etc.)
- Code-transport / -serialization
- Visualize algorithms as you would visualize data
- Consider application on code for every algorithm developed for data handling.

# Data as Code

Analyse and treat code, identically to data

- adapt, configure, produce
- mutate, optimize, fix
- compression (lossless...?)
- develop

# Why „bionic“?



Equivalent to how biological systems are build from smaller, highly optimized interconnected sub-systems.

Systems can adapt to changing requirements.

# Thank you

## PantheOS.net e.V.

„Research and distribution of [such] software technology.“

## Modular Mind Labs GmbH

Contact with customers in context of cloud projects.

Plan: full Open Source access and openness for community

Work in rapid progress. Stay tuned: <https://pantheos.net>  
(don't be disappointed, when you click today. Patience...)