HS^3 - A serialization standard for statistical models in high energy physics

Carsten Burgard[1] , Cornelius Grunwald[1] , Robin Pelkner[1] , Oliver Schulz[2]

8th BCD ISHEP Cargèse School 29.03.2023

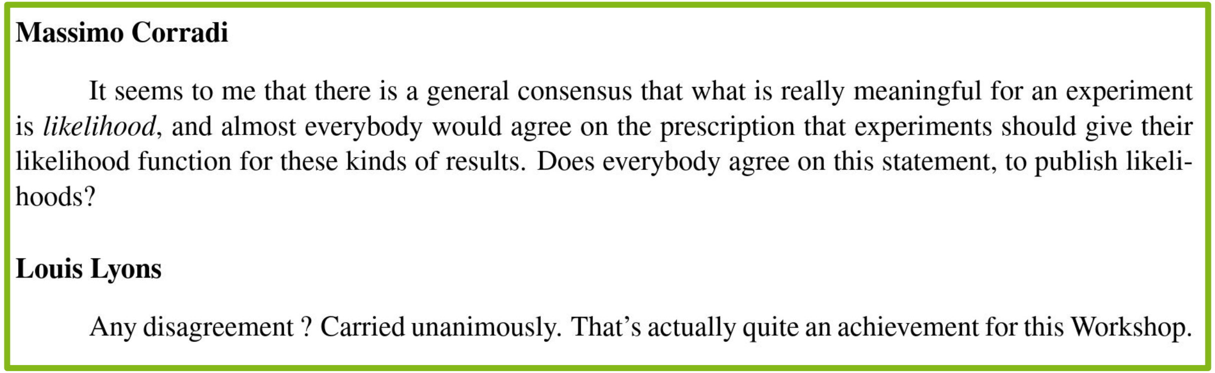
[1] TU Dortmund University, AG Kröninger
[2] Max Planck Institute for Physics, Munich

technische universität
dortmund

The open science approach

- open science: publish results and data
- statistical models are necessary:
  - validation and reproduction of results
  - reinterpretation and combination
  - publication and archiving

  } good scientific practice
- 1st Workshop on Confidence Levels 2000:



Massimo Corradi

It seems to me that there is a general consensus that what is really meaningful for an experiment is *likelihood*, and almost everybody would agree on the prescription that experiments should give their likelihood function for these kinds of results. Does everybody agree on this statement, to publish likelihoods?

Louis Lyons

Any disagreement ? Carried unanimously. That's actually quite an achievement for this Workshop.

→ experiments should publish likelihoods

Current status in HEP

- ROOT stores statistical models (RooWorkspace) in binary format (".root" files)
- pyhf stores HistFactory-Models in human-readable JSON-Files
  - already used in many analyses
  - BUT: restrained to HistFactory-like models
- there is other statistical software, models and tools, but no standardized format

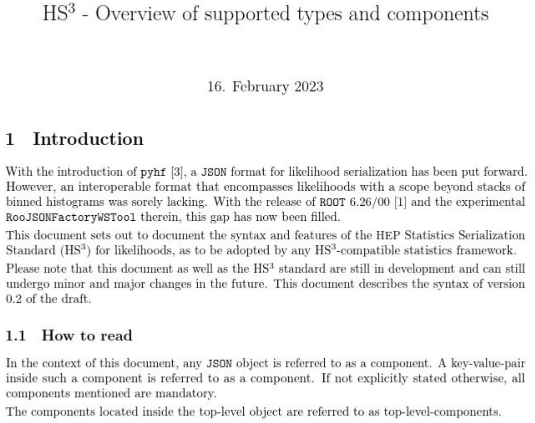
HS^3 - HEP Statistics Serialization Standard

technische universität
dortmund

idea: provide standardized format for statistical models:

- human-readable, in JSON format
- machine-readable for direct implementation of statistical models
- software-independent
- generic, mathematical definitions
- full compatibility with respect to RooWorkspace and pyhf

https://github.com/hep-statistics-serialization-standard



HS^3 - Overview of supported types and components

16. February 2023

1 Introduction

With the introduction of pyhf [3], a JSON format for likelihood serialization has been put forward. However, an interoperable format that encompasses likelihoods with a scope beyond stacks of binned histograms was sorely lacking. With the release of ROOT 6.26/00 [1] and the experimental RooJSONFactoryWSTool therein, this gap has now been filled.

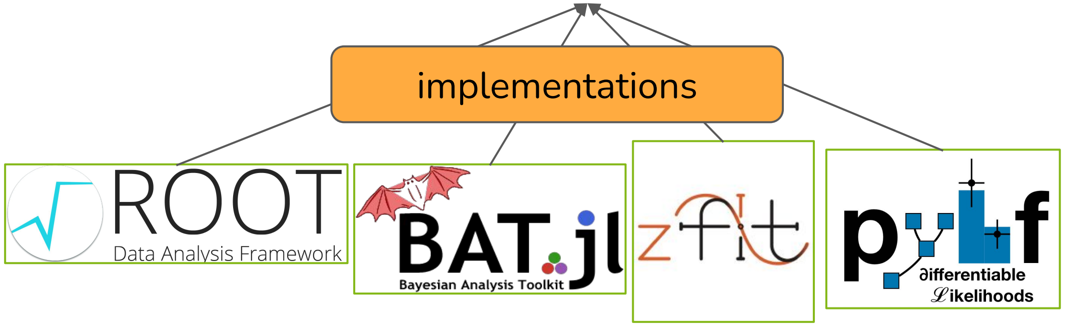
This document sets out to document the syntax and features of the HEP Statistics Serialization Standard (HS^3) for likelihoods, as to be adopted by any HS^3-compatible statistics framework.

Please note that this document as well as the HS^3 standard are still in development and can still undergo minor and major changes in the future. This document describes the syntax of version 0.2 of the draft.

1.1 How to read

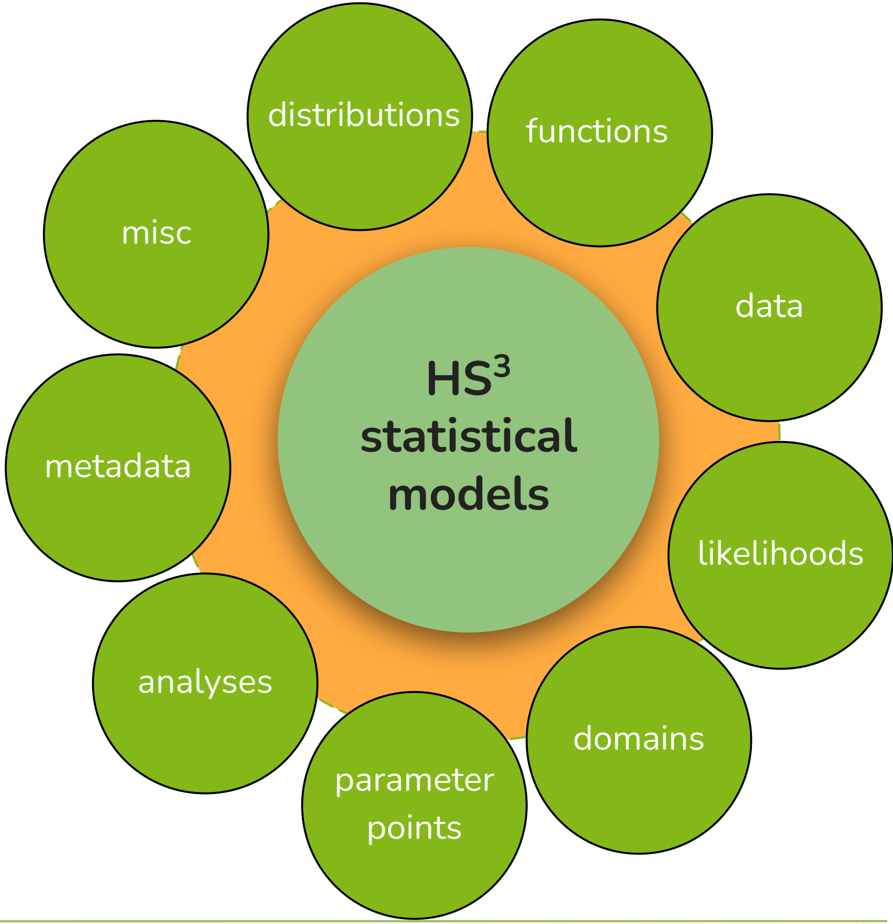
In the context of this document, any JSON object is referred to as a component. A key-value-pair inside such a component is referred to as a component. If not explicitly stated otherwise, all components mentioned are mandatory.

The components located inside the top-level object are referred to as top-level-components.



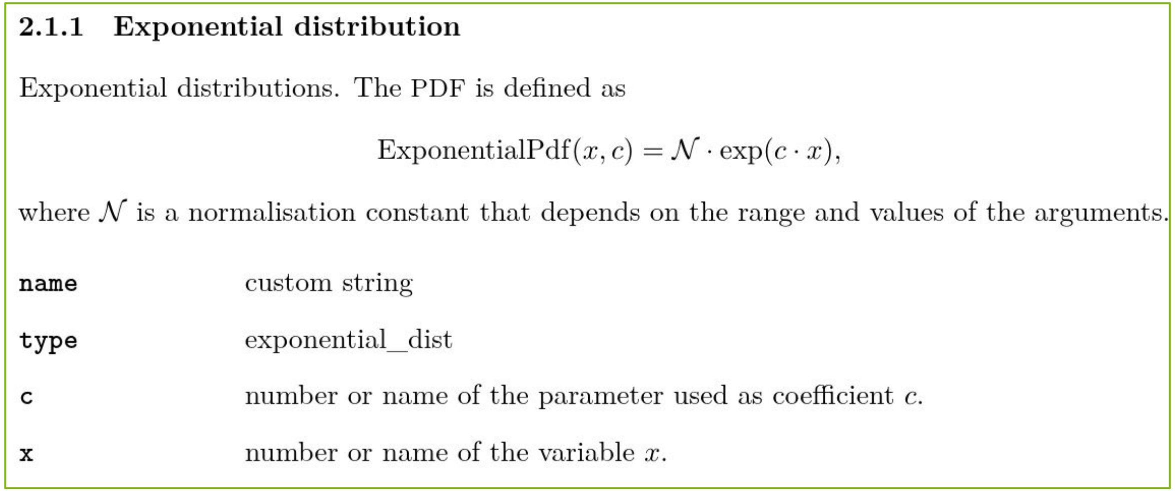
Top-level elements

- HS^3 includes everything needed for a complete representation of an analysis
- flat structure of elements each accessible on their own
- every element is completely optional depending on the model
- the elements can depend on each other



Distributions

- objects with unique name, type of distribution and respective parameters



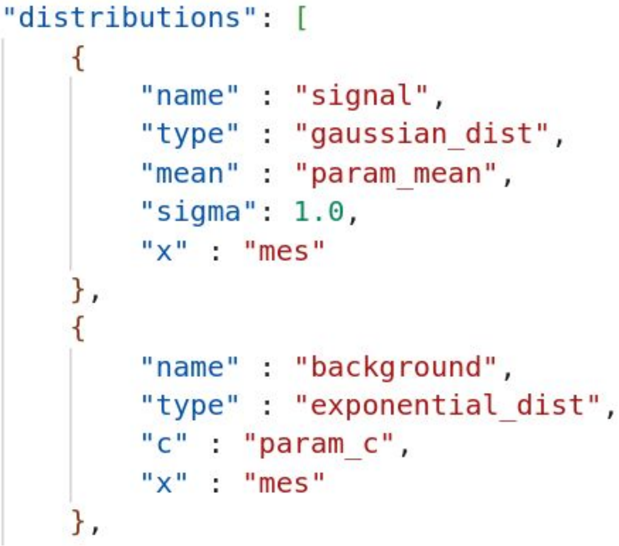
2.1.1 Exponential distribution

Exponential distributions. The PDF is defined as

$$\mathrm{ExponentialPdf}(x, c) = \mathcal{N} \cdot \exp(c \cdot x),$$

where $\mathcal{N}$ is a normalisation constant that depends on the range and values of the arguments.

name	custom string
type	exponential_dist
c	number or name of the parameter used as coefficient c.
x	number or name of the variable x.



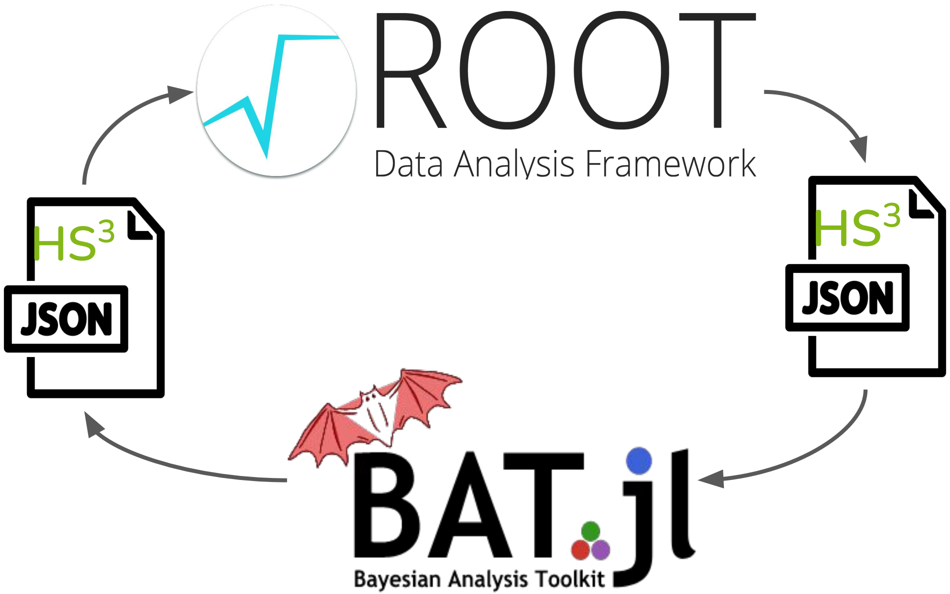
```
"distributions": [
    {
        "name" : "signal",
        "type" : "gaussian_dist",
        "mean" : "param_mean",
        "sigma": 1.0,
        "x" : "mes"
    },
    {
        "name" : "background",
        "type" : "exponential_dist",
        "c" : "param_c",
        "x" : "mes"
    },
```

- further distributions include:

  HistFactory channel, Gaussian, Poisson, Polynomial, Mixture Distribution, Product Distribution ... and growing!

Current status

- Currently ongoing implementations in ROOT and Julia (BAT.jl) of HS^3
- Idea: first round trip
- ongoing full harmonization with pyhf
- zfit: included in meetings and ongoing discussions

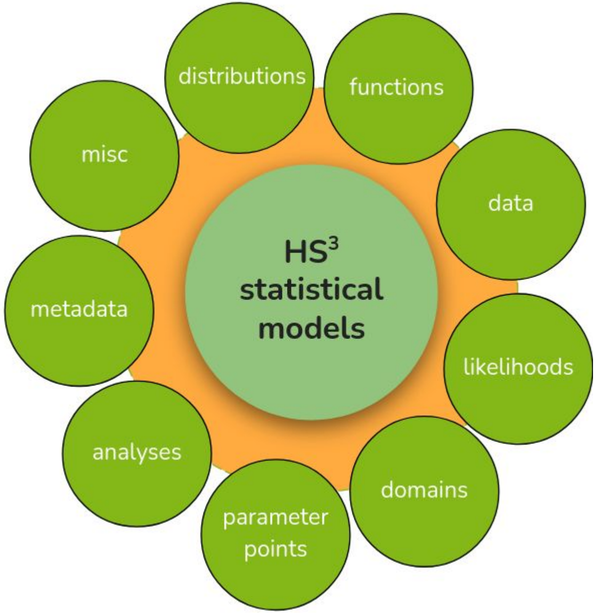


zfit

pyhf
∂ifferentiable
ℒikelihoods

Conclusion and outlook

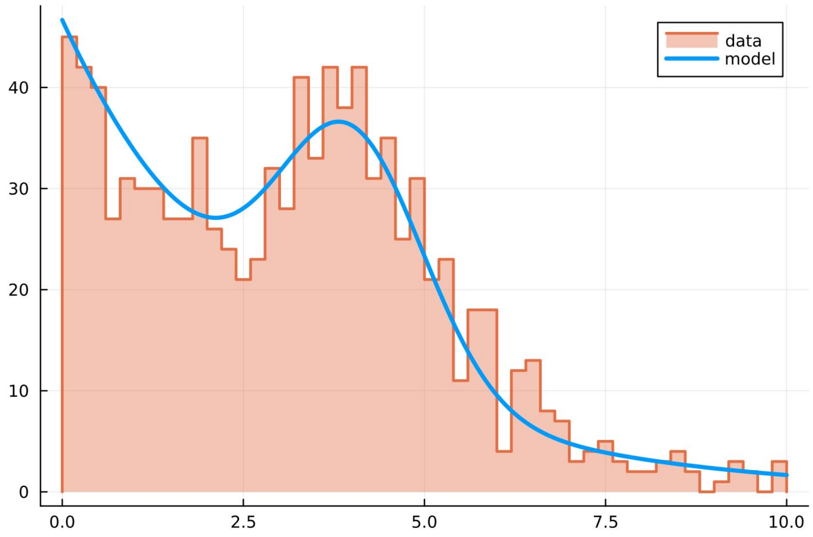
- HS^3 is an evolving standardized way of distributing statistical models
- preliminary HS^3 version implemented since ROOT 6.26
- release of this HS^3 version in RooFit Update this summer
- currently working on roundtrip between ROOT & BAT.jl
- part of the IRIS-HEP Strategic Plan for the Next Phase of Software Upgrades for HL-LHC Physics
- first complete version by the end of 2023



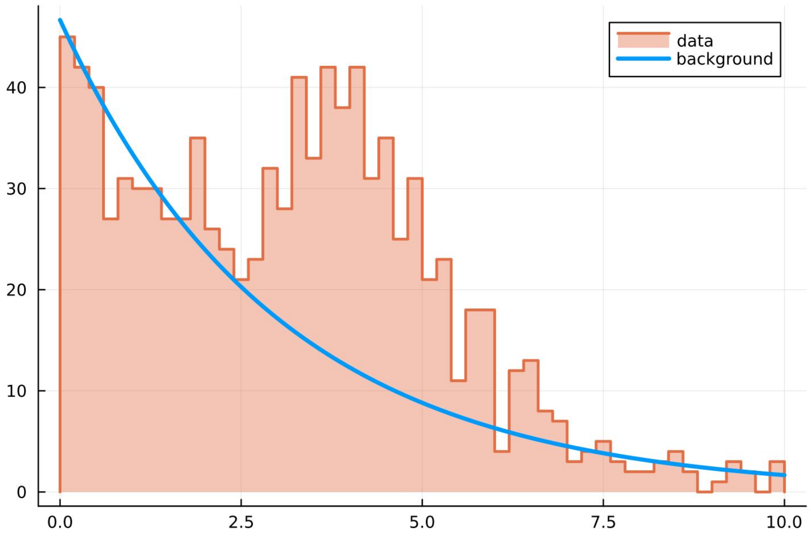
Backup

Simple example

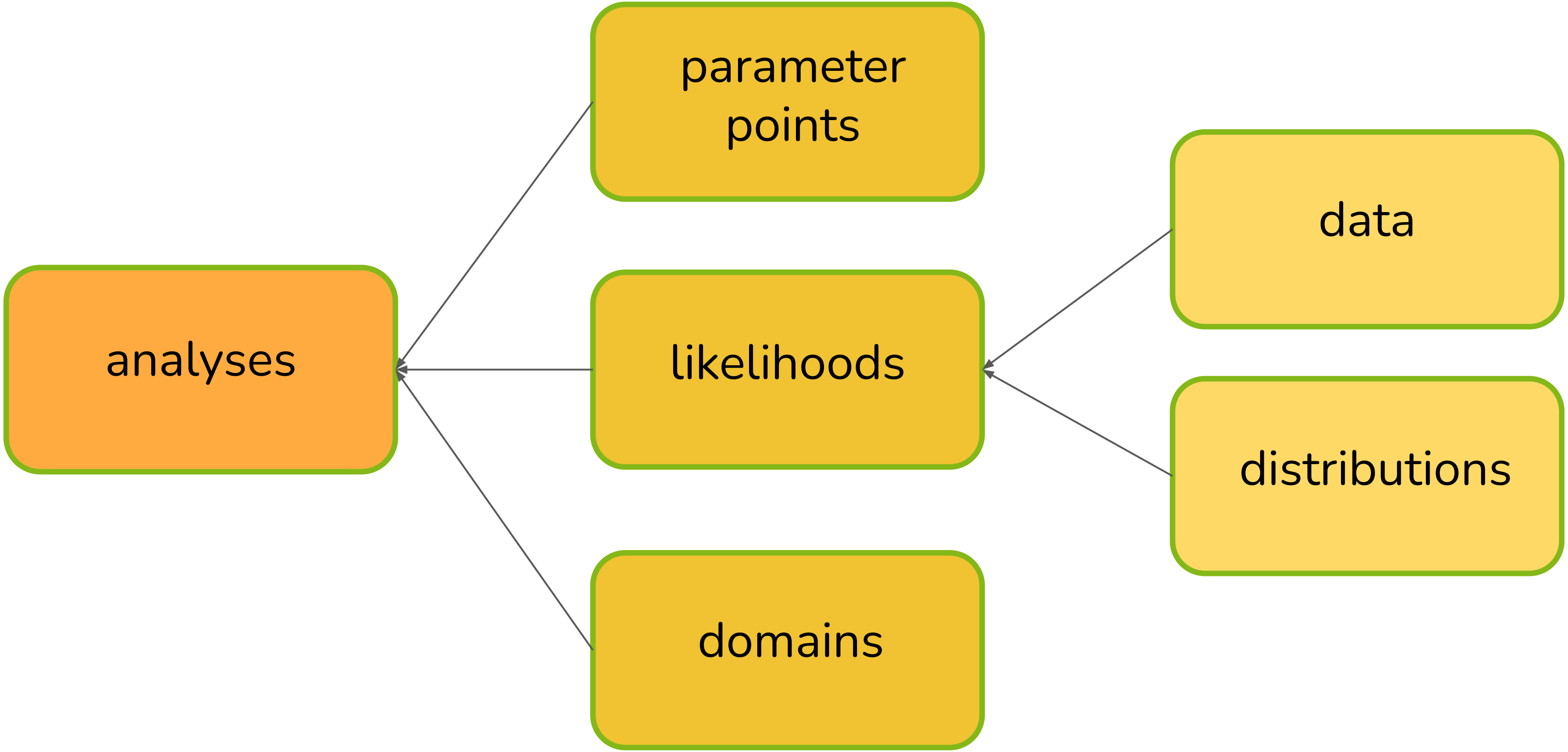
signal with exponential background



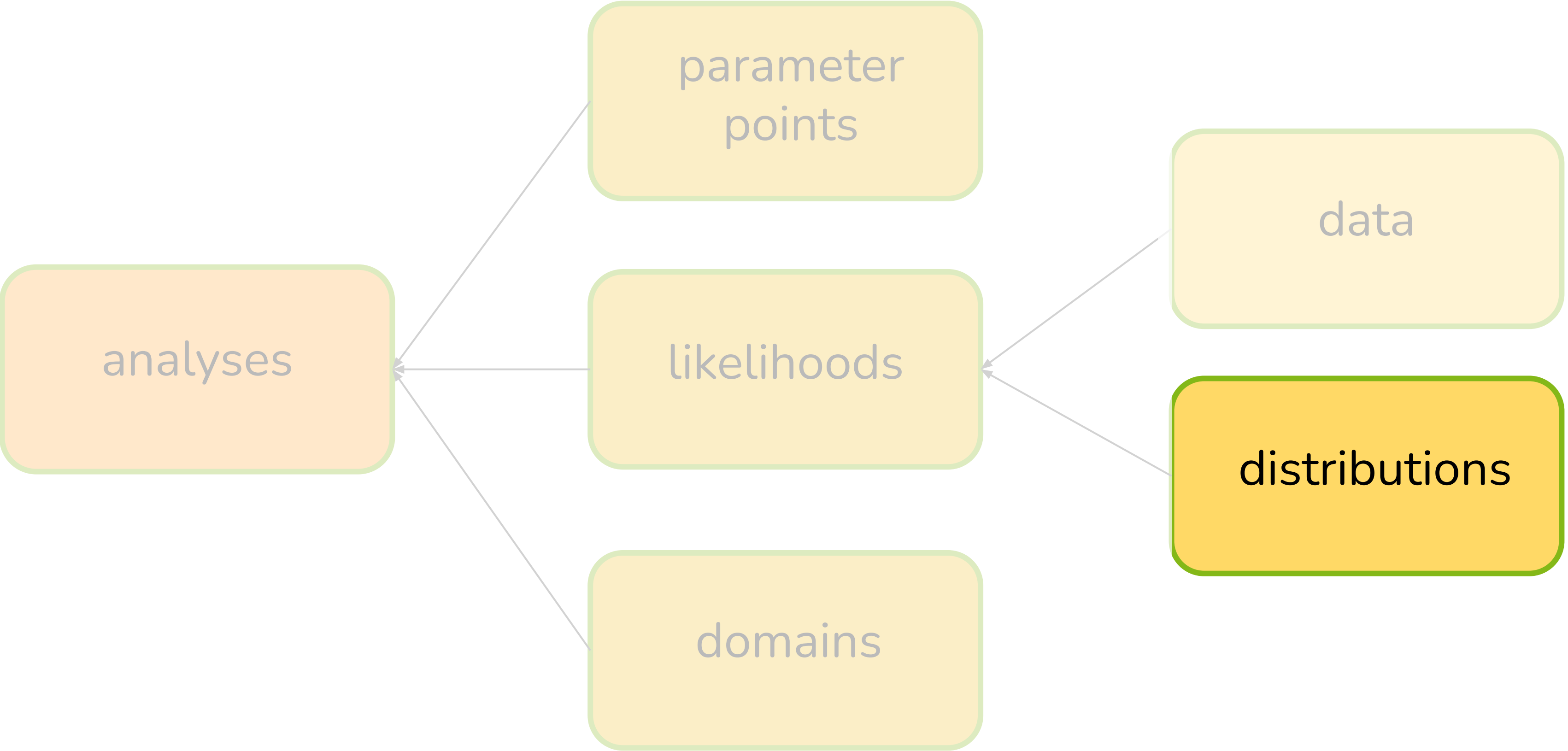
background only fit



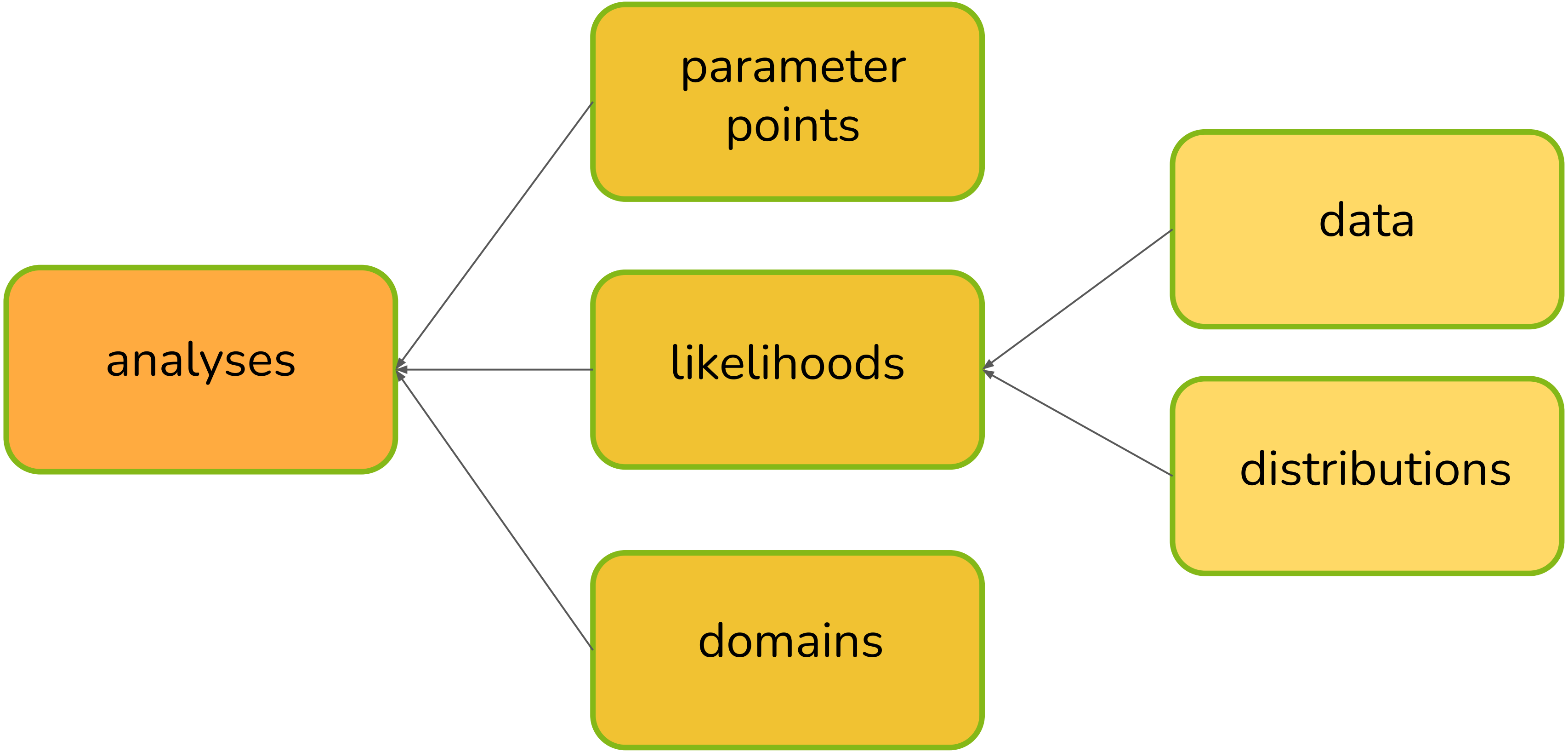
HS^3 - Structure



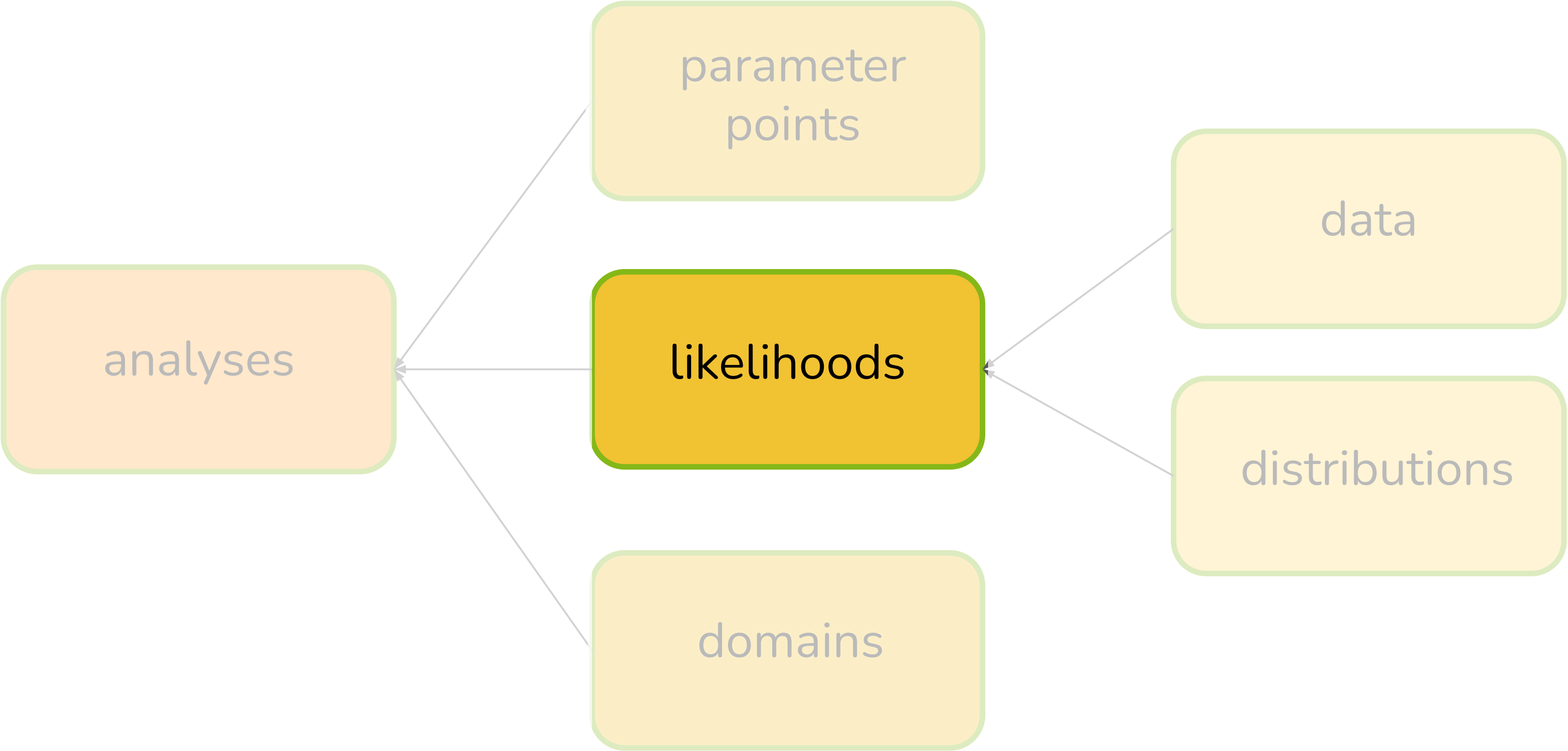
A more detailed view



HS^3 - Structure



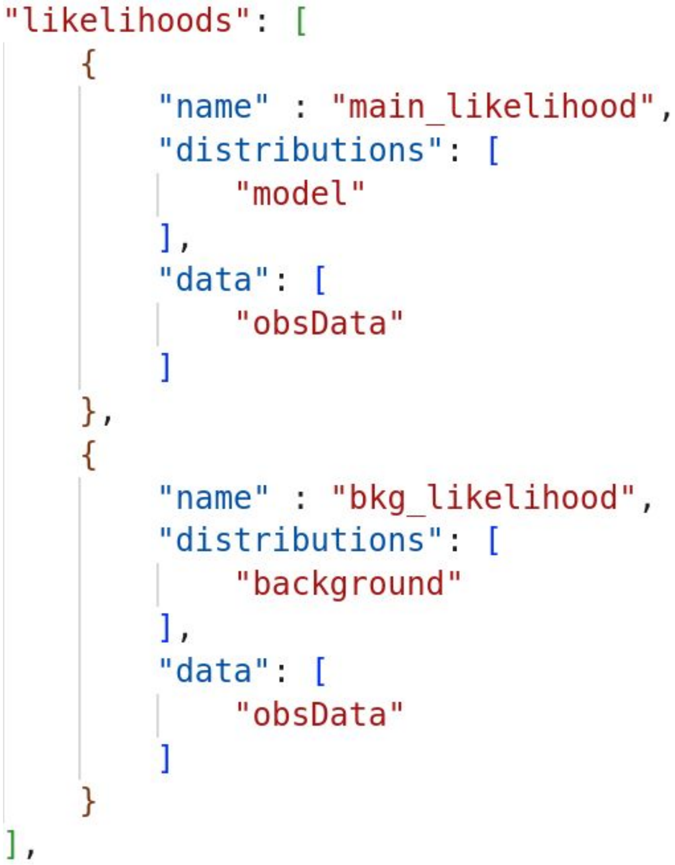
Bringing it together...



Likelihoods

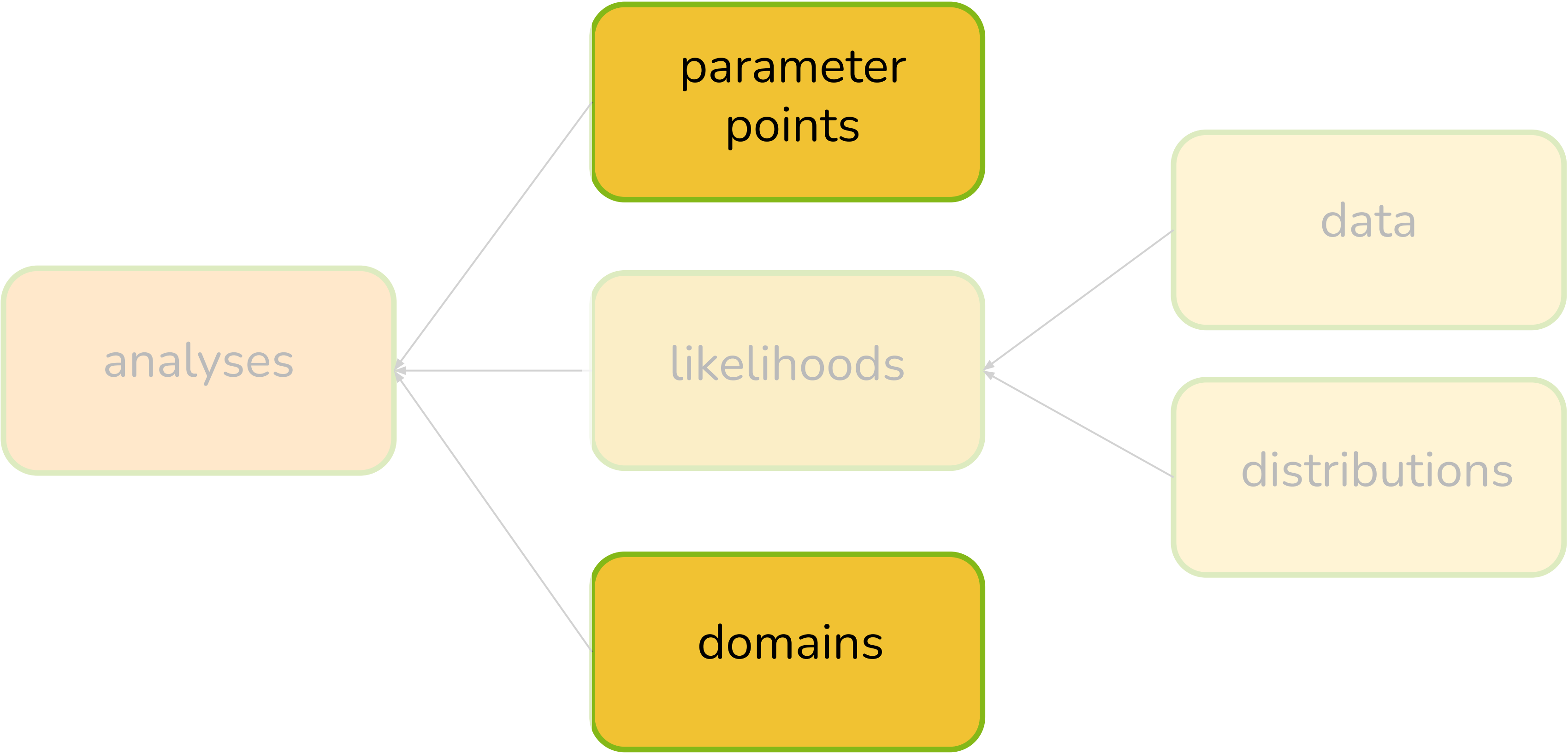
- Combination of data and distributions to likelihoods

- multiple likelihoods can be defined for multiple analyses, e.g.
  - background only fit
  - background + signal fit



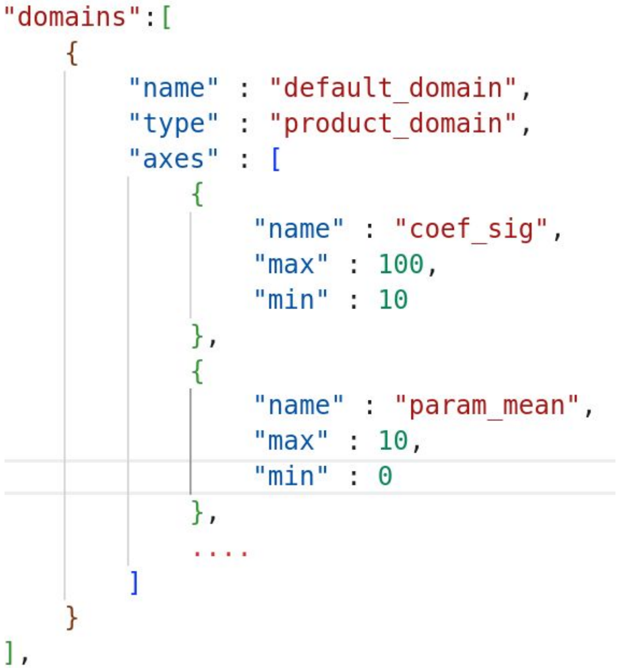
```
"likelihoods": [
    {
        "name" : "main_likelihood",
        "distributions": [
            "model"
        ],
        "data": [
            "obsData"
        ]
    },
    {
        "name" : "bkg_likelihood",
        "distributions": [
            "background"
        ],
        "data": [
            "obsData"
        ]
    }
],
```

Bringing it together...



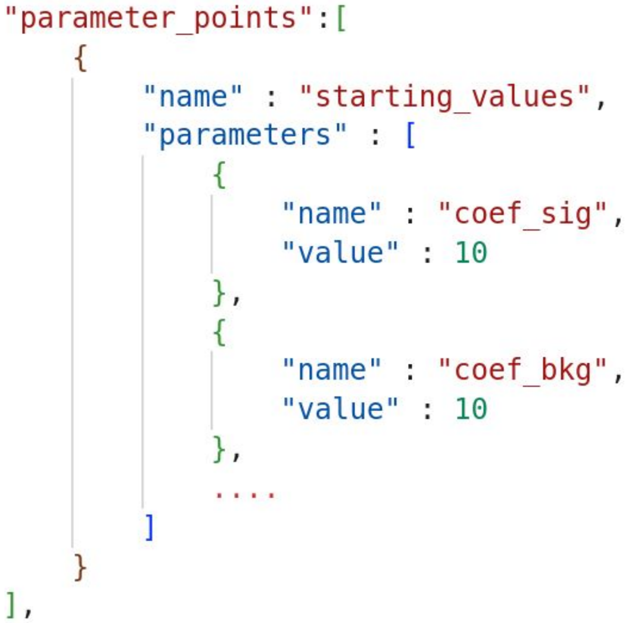
Domains and parameter points

domains: ranges of parameters



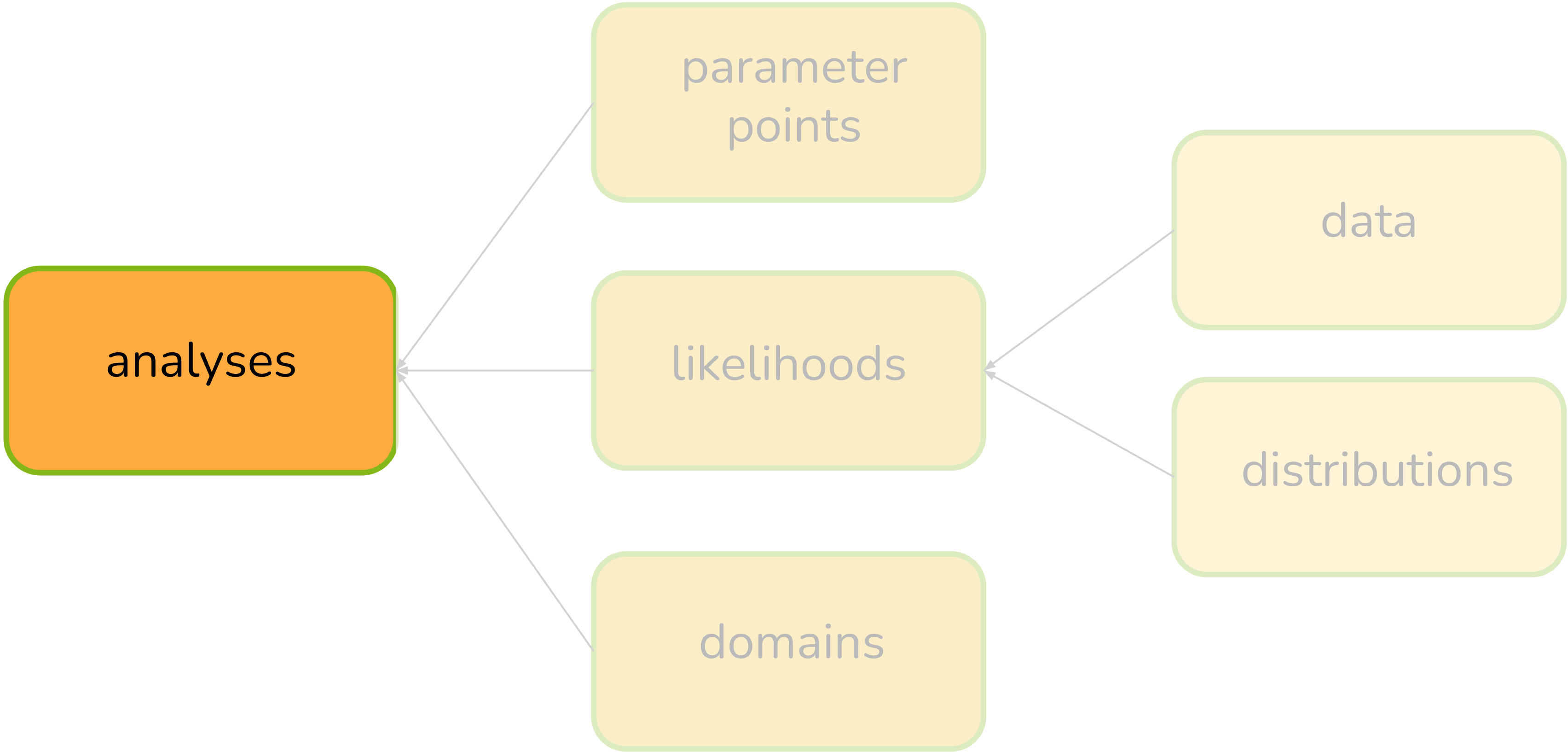
```
"domains":[
    {
        "name" : "default_domain",
        "type" : "product_domain",
        "axes" : [
            {
                "name" : "coef_sig",
                "max" : 100,
                "min" : 10
            },
            {
                "name" : "param_mean",
                "max" : 10,
                "min" : 0
            },
            ....
        ]
    }
],
```

parameter points: estimates, parameter settings, best-fit values, starting values....



```
"parameter_points":[
    {
        "name" : "starting_values",
        "parameters" : [
            {
                "name" : "coef_sig",
                "value" : 10
            },
            {
                "name" : "coef_bkg",
                "value" : 10
            },
            ....
        ]
    }
],
```

Bringing everything together



analyses

parameter points

likelihoods

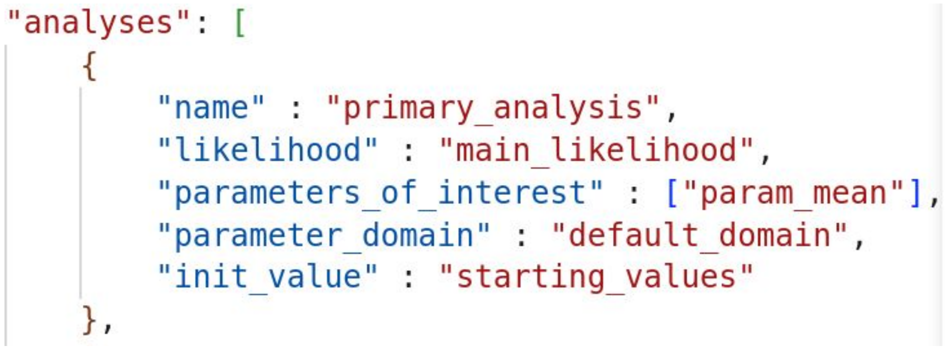
domains

data

distributions

Analyses

- combines all previously defined elements
- allows to automate full analyses



```
"analyses": [
    {
        "name" : "primary_analysis",
        "likelihood" : "main_likelihood",
        "parameters_of_interest" : ["param_mean"],
        "parameter_domain" : "default_domain",
        "init_value" : "starting_values"
    },
```

Is that enough for every possible analysis?

- HS^3 provides more options to define, e.g., auxiliary likelihoods, parameter estimates, and prior distributions for Bayesian analyses
- for more details see: https://github.com/hep-statistics-serialization-standard