technische universität dortmund

SFB 876 Providing Information by Resource-Constrained Data Analysis

lehrstuhl physik e5

Corsika in C++

Dominik Baack

Next Gen Corsika Workshop

Karlsruhe

24.06.2018

Myself

- Dominik Baack
- Technical University Dortmund
- Responsible for MC Production in Magic (+Fact)
- Developer of Cor-PlusPlus extensions:

  Dynstack-Extended RemoteControl ParallelChernekov

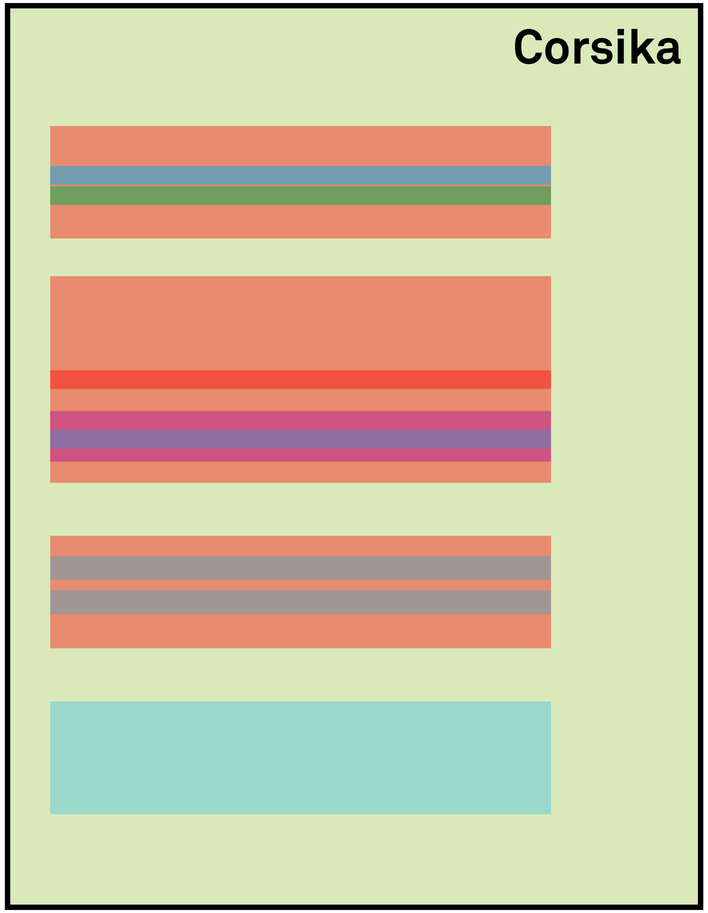
Target Goal

- Speed/performance increase in Corsika
- Better accessibility and development in Corsika
- Test of „new“ patterns and structure

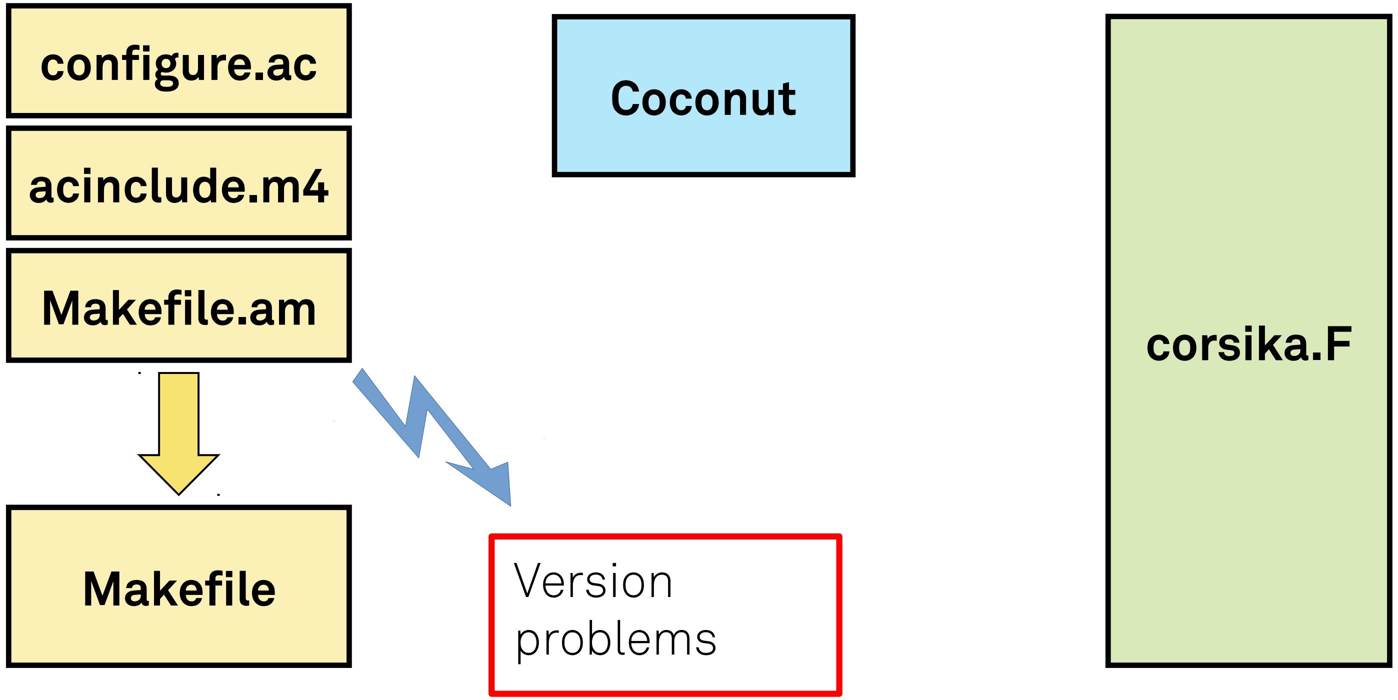
Insights gained during development

Internal structure

- Monolithic application
- Nested structure makes it difficult to maintain
- Nearly each function has side-effects !!
- New modules need modifications of existing code in multiple positions



Current workflow



Compile Time Modules

- Deployment:

Compilation
• Templates possible • High performance (full Compiler optimization [-O3])
• Inflexible • Increasing size of main module

Link Time Modules

- Deployment:

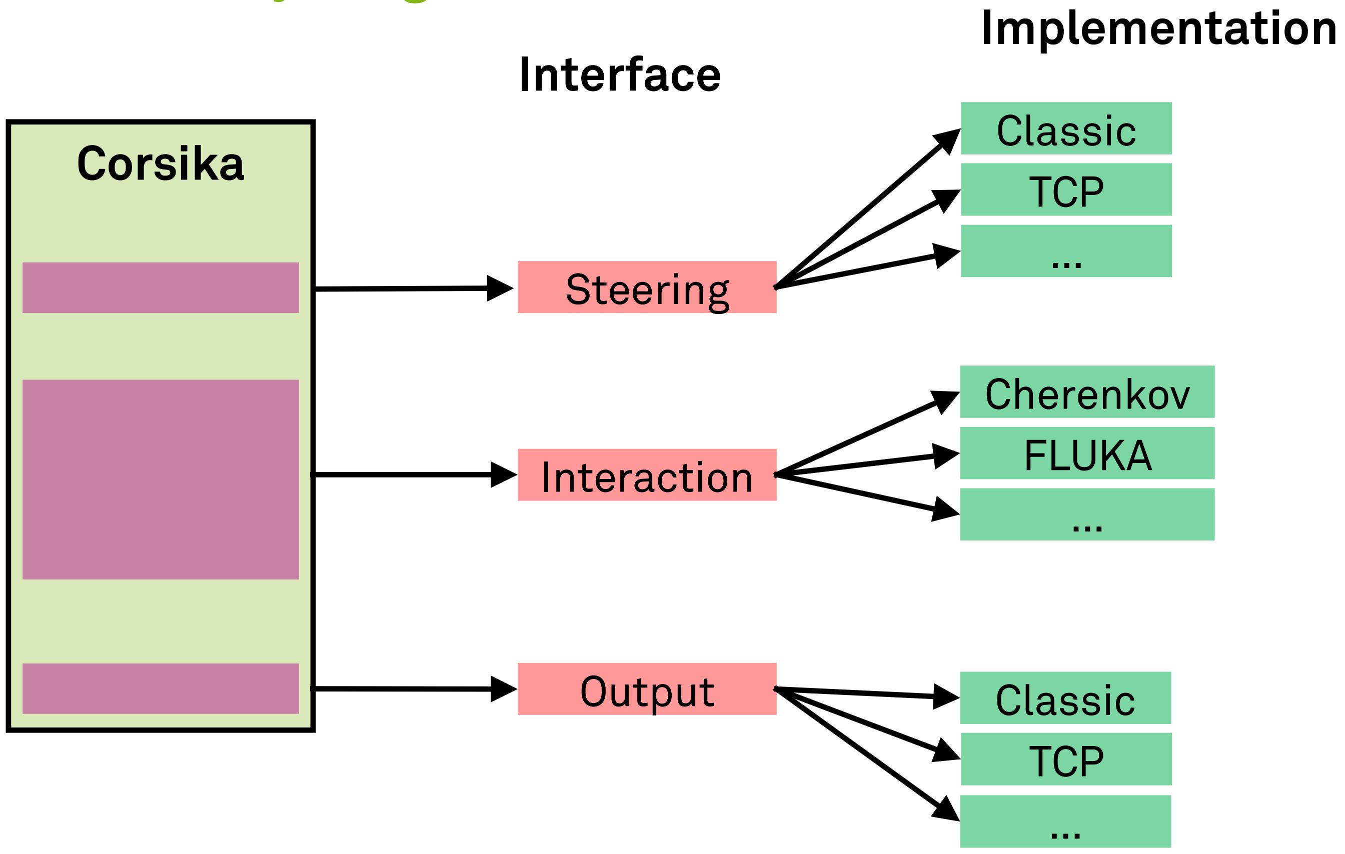
Static linking
• Modular development • Separate tests • Good performance (linker optimization)
• Only flexible at compile time

Run Time Modules

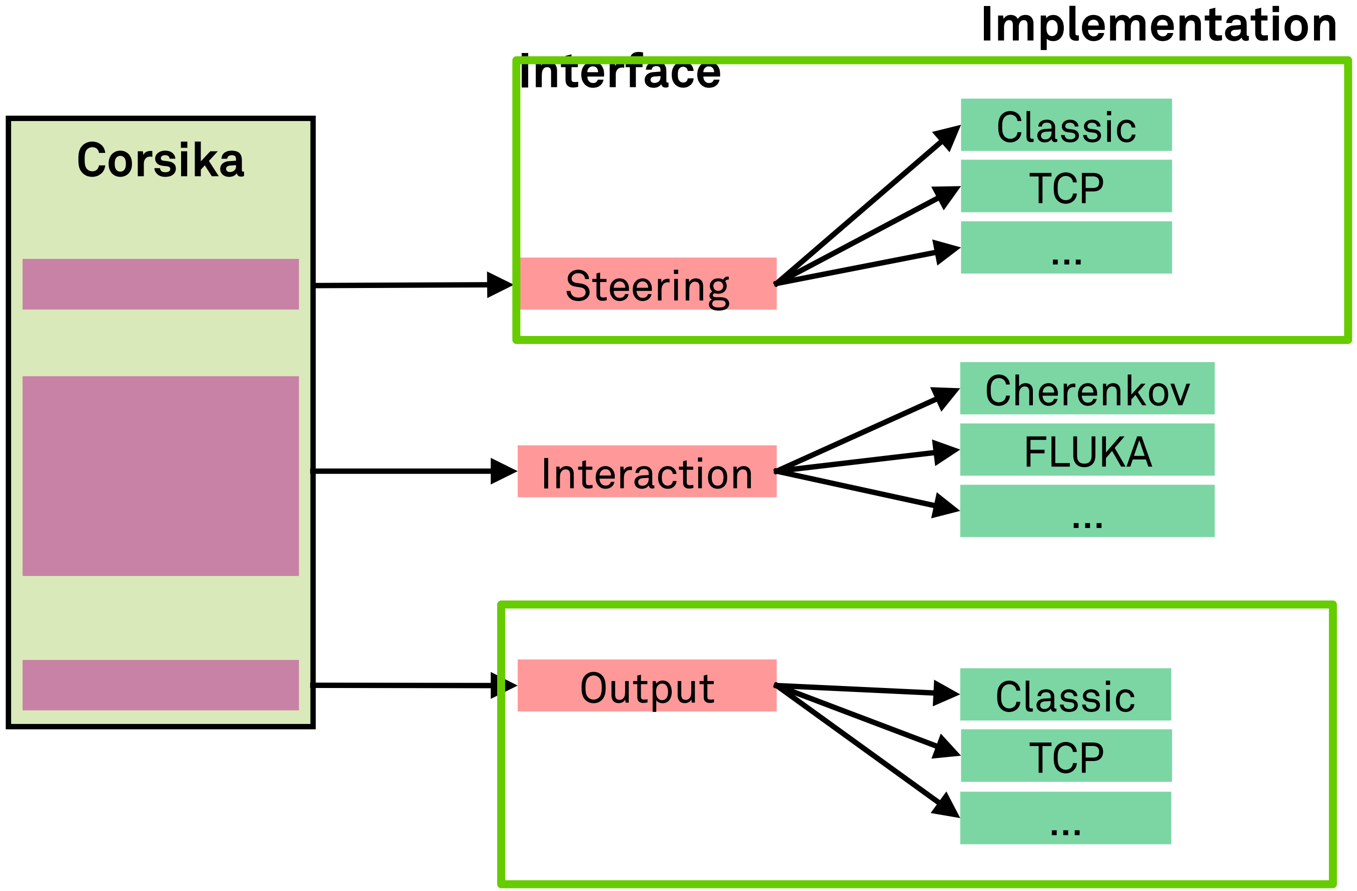
- Deployment:

Dynamic linking
• very flexible • Precompiled distribution (closed source)
• Lowest performance

Interface Everything



Interface Everything



Runtime Polymorphism

- Virtual is not evil by default!
  Depending on the implementation virtual has close to zero impact on performance
- Virtual allows additional variability and type safety during development
- Easy interchangeable for rapid prototyping → not performance critically
- Strict guidelines for external developer

Runtime Plugins → SharedObjects

- Plug-in-System, based on shared libraries, for parts with low call frequency (0 to 100Hz)
- Fast exchange and completely independent development possible
- Closed Source possible
- Complete different languages possible

Exceptions

- Use of exceptions is highly encourage to improve debugging
- Memory Dump and controlled shutdown can prevent data loss and faster debugging
- Exception implementation of current compiler have Zero Costs, but no option for usual flow control