



Hierarchy Software Development Framework (h-dp-fwk) Project

Alexander Zaytsev // Alexander.S.Zaytsev@gmail.com

Budker Institute of Nuclear Physics (BINP)

Novosibirsk, Russia

CHEP2009 (Prague)



Outline

- Introduction
- Project architecture
- Key design assumptions
- Review of the main features
- Some details on implementation
- **Summary and future plans**



Introduction to the Project

- Hierarchy Software Development Framework (h-dp-fwk) is a lightweight tool for building portable modular applications for performing data analysis in a batch mode.
- Initially the project was targeting the case of building experimental data processing applications for **CMD-3 experiment (BINP)** and then it was redesigned from the scratch in order to match the generic case of modular data processing application operating within the well defined computing environment.
- The licensing schema for the source code, binaries and documentation implies that the product is free for non-commercial use.
- Official website: <http://hdpfwk.org>



Key Design Assumptions

- The target application should be easily decomposable into a set of modules while the rate of switching control from one module to another is within the sequence is relatively low thus the performance overhead introduced by the Framework itself is negligible (HEP offline event processing is the ideal case of such conditions)
- Direct interactions between modules and non-Framework based session control/modules configuration are avoided whenever it is possible
- The application runs in a batch environment thus no interactive features needed (CLI tools only)



Pros & Cons of Relying on h-dp-fwk Framework

- What developers and users gain (**pros**):
 - Well formed logging, debugging and runtime consistency verification tools
 - XML configuration handling tools with internal consistency verification blocking parsing mechanisms when needed
 - Data processing session control interfaces (CLI/API)
- What they ought to accept (**cons**):
 - Studying new interfaces and tools
 - Getting used to the CLI/API, XML configuration and data processing schema proposed (basic knowledge of XML, XML Schema & XInclude concepts is required)
 - Design the applications with respect to the Framework' architecture



Pros & Cons of Relying on h-dp-fwk Framework

- (pros):
 - Message and data exchange tools
 - Dependency resolution mechanisms for both forward and reversed lookup
 - No need to spend resources on building your own runtime environment, so you can concentrate on the nontrivial parts of the code which matters most
- (cons):
 - Getting used to modules interaction schema in the reverse dependency lookup mode (DAG)
 - The application must be large enough and significantly long living in order to benefit from using the mechanisms provided by the Framework
 - Low performance overheads for applications with intensive message exchange is not guaranteed



Main Features

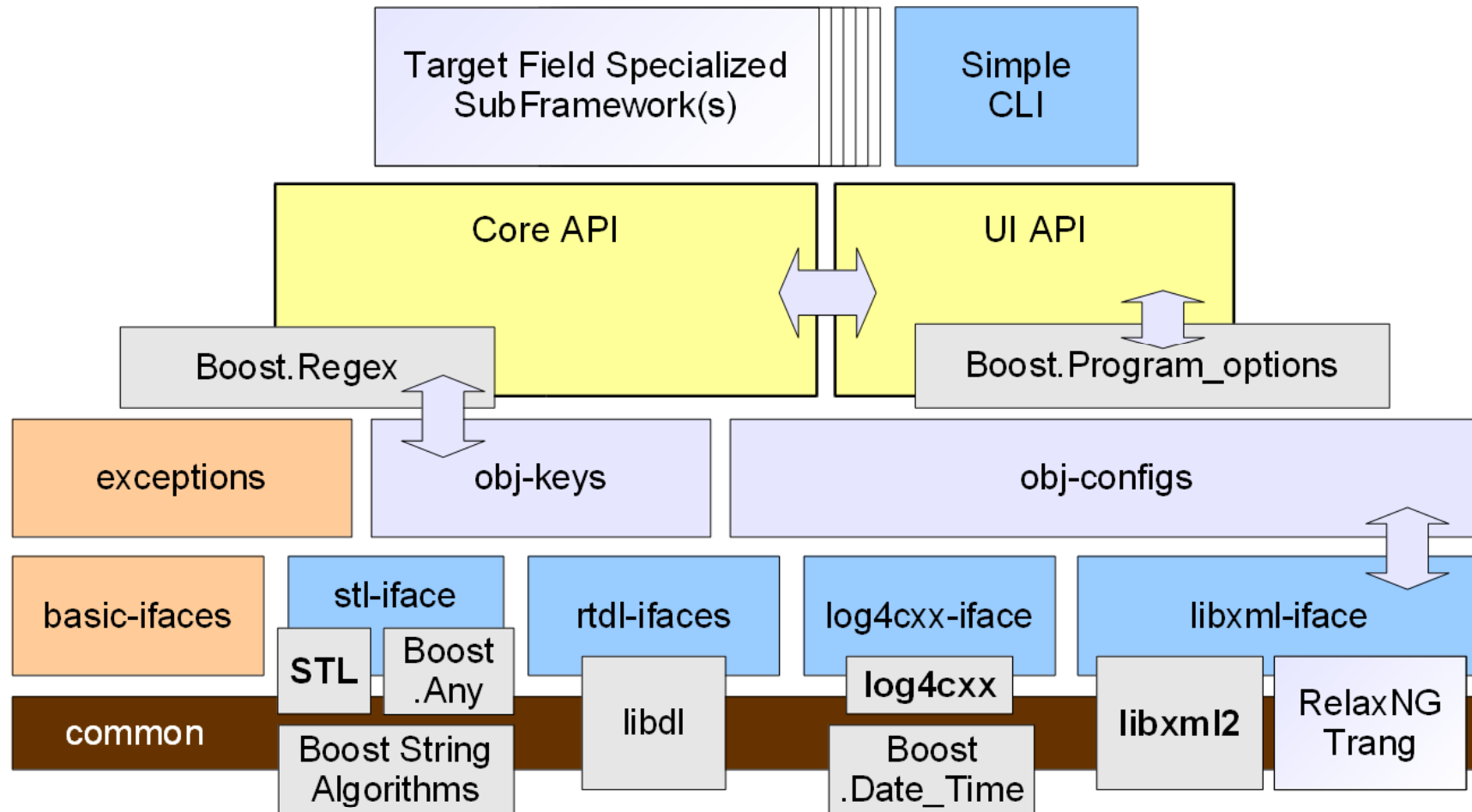
The challenge is to build the system which is suitable for both development and production environment.

The main features of the Framework are:

- Modularity,
- Built-in message and data exchange mechanisms,
- XInclude and Schema enabled XML configuration management tools,
- Dedicated log management tools, internal debugging tools,
- Both dynamic and static module chains support,
- Internal DSO version and consistency checking,
- SCons-base build and packaging system for handling both core components and modules,
- Well defined API for creating specialized sub-frameworks.

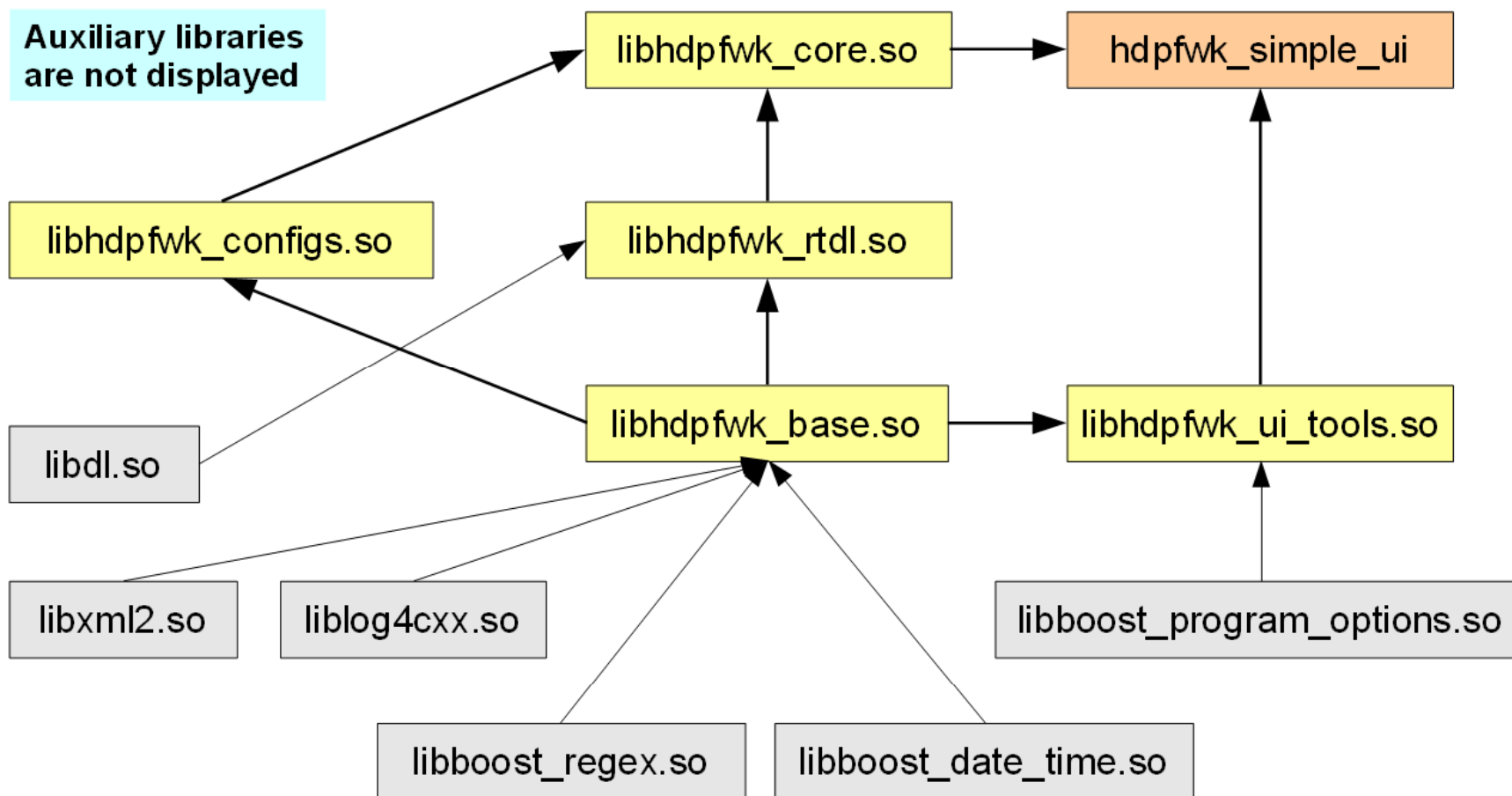


Project Architecture Layout





External Dependencies



Static linking is also supported (without RTDL interfaces)

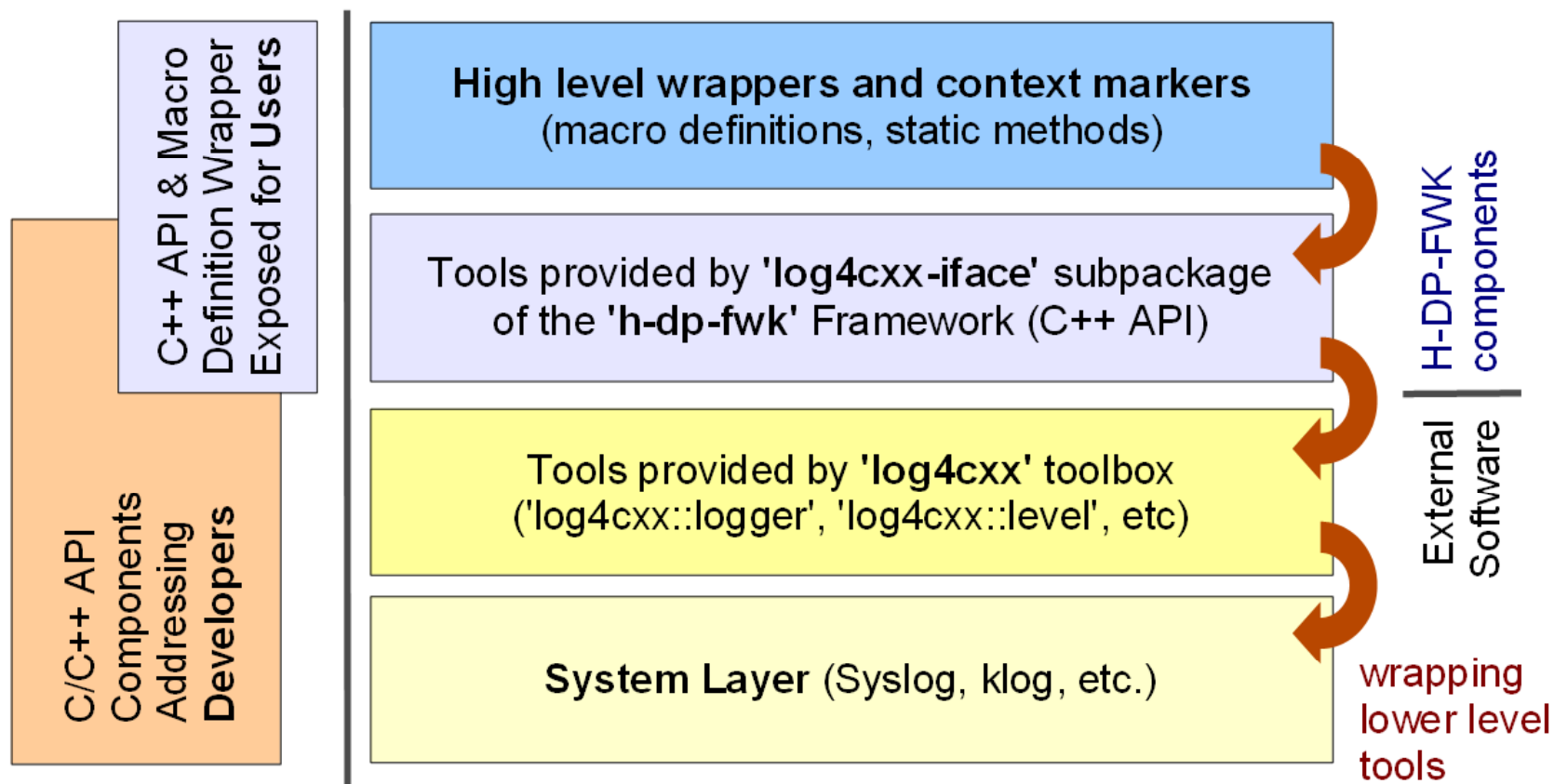


Message Management Tools

- Centralized logger of the Framework is implemented on top of the **log4cxx** toolbox which is capable of handling named logger hierarchies and provide a variety of message I/O and filtering tools
- **Use of high level macro wrappers allows one to remove all the logger related code or just a part of it via build system interface without changing the code**
- Logging and debugging tools are the most fundamental part of the Framework:
 - Almost all the project components depend on
 - The message handling schema implemented is intended to be used not only by the base Framework itself, but by user modules and derived Frameworks as well
 - **Internal control message mechanism involved is base entirely on exceptions**



Logger Architecture Overview



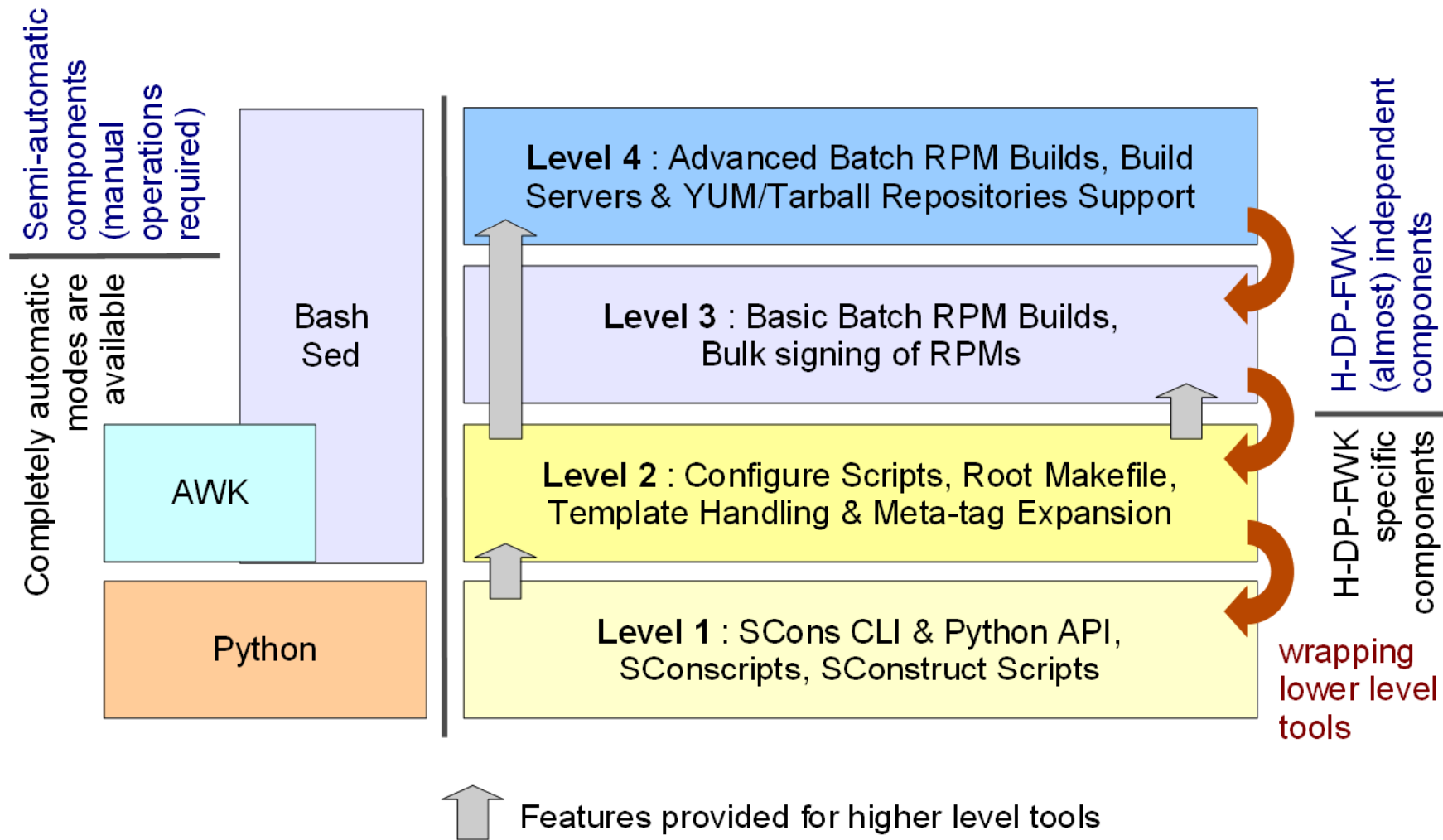


Build System

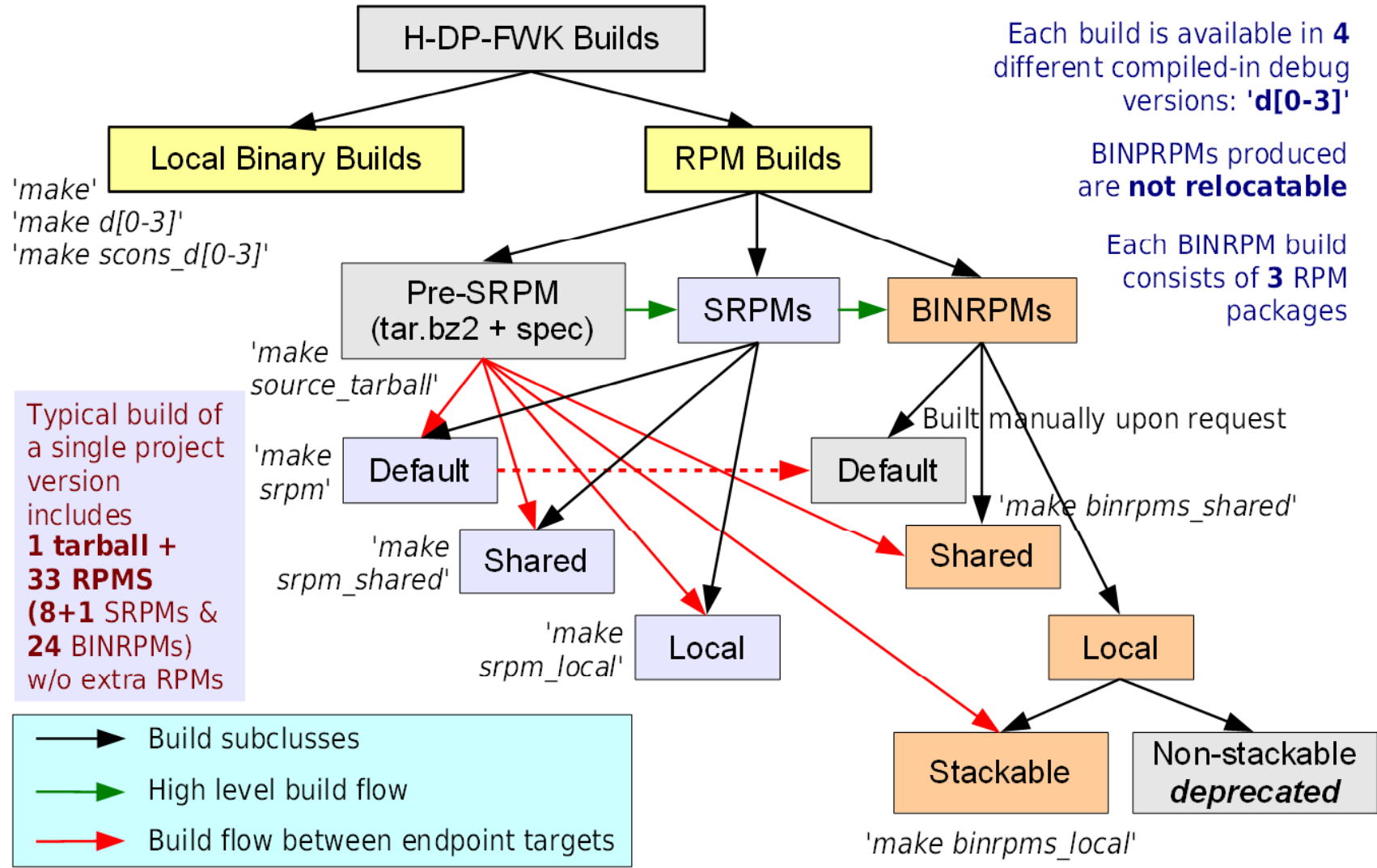
- Build system exploited by the project is based on **SCons** product which is an open source Python based multi-platform build system and high level wrappers placed on top of it
- It is aware of high level logger tuning options and capable of producing the binaries with requested level of verbosity selected
- In addition it supports source code tree management tools (cleaning, spellchecking, Doxygen documentation production), **batch RPM/tarball builds** and provides a simple implementation of the **build server functionality**



Build System Architecture Summary

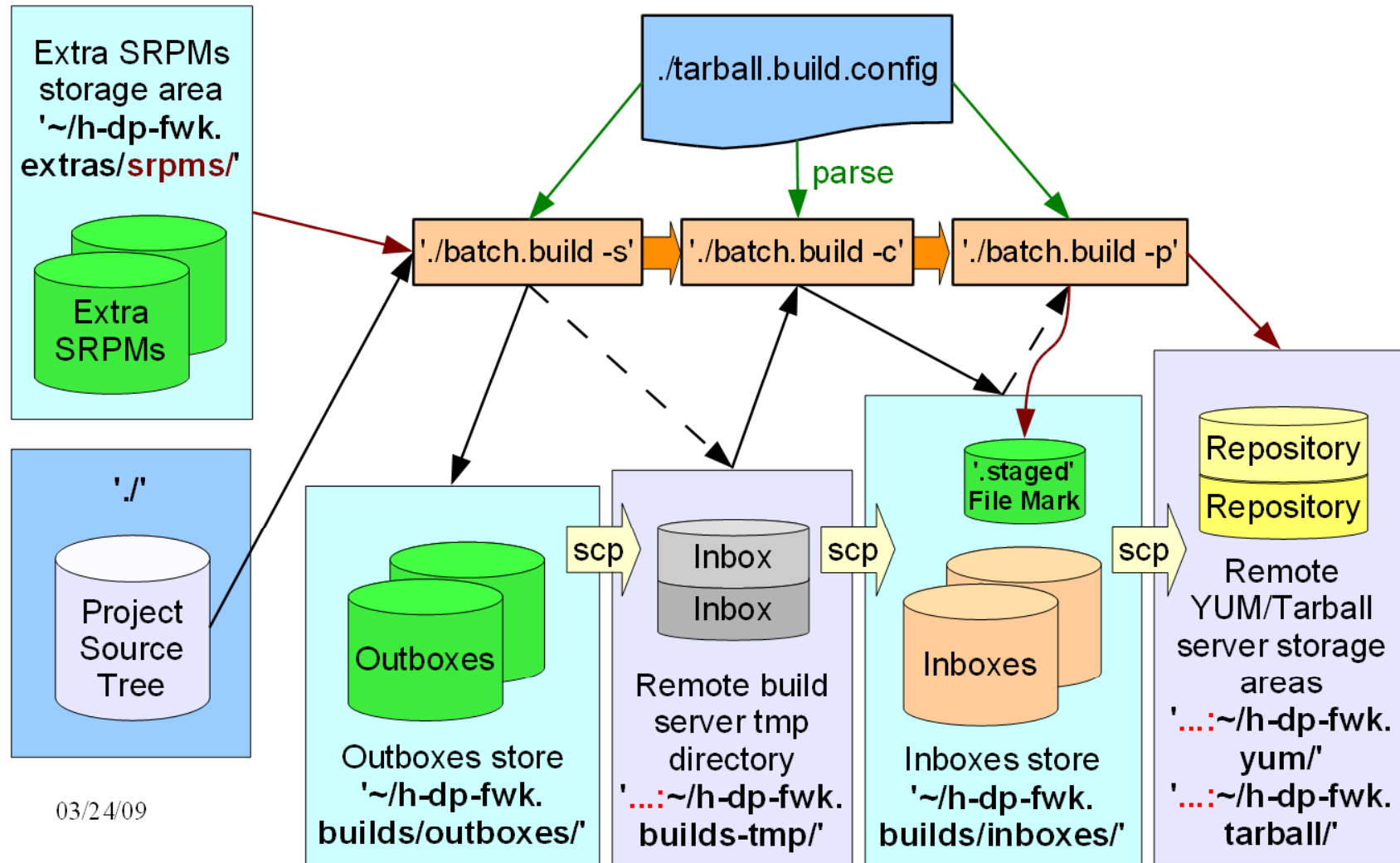


Variety of Builds Supported





Build Servers Support





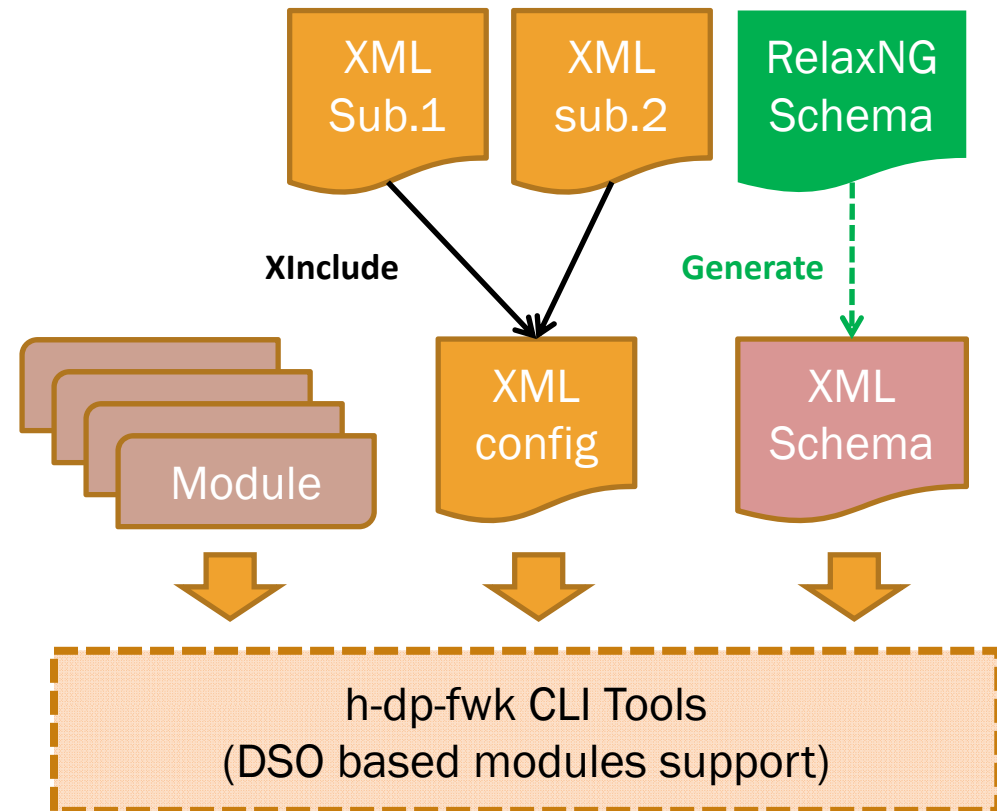
Data Processing Session

- Data processing schema is defined by the selected approach to inter-modules interaction while wide spectra of solutions are supported:
 - Hard-coded sequence (*direct module chain resolution, sequence is defined by XML configuration*)
 - DAG sequence resolution (*reverse module chain resolution, sequence can be defined by both module's code and XML configuration*)
 - Supervisor modules control (*arbitrary module chain resolution defined mostly by the supervisor module's code*)
 - **Mixture of solutions listed above**



XML Configuration Handling

- **libxml2** is used as a low level handler of configuration files
- High level wrappers provide **XInclude/XML Schema** support and a user tunable XML configuration subsections override mechanism
- Combined these two features deliver a significant increase of usability when it comes to dealing with the large configuration (e.g. global reconstruction)





Data Access Mechanism

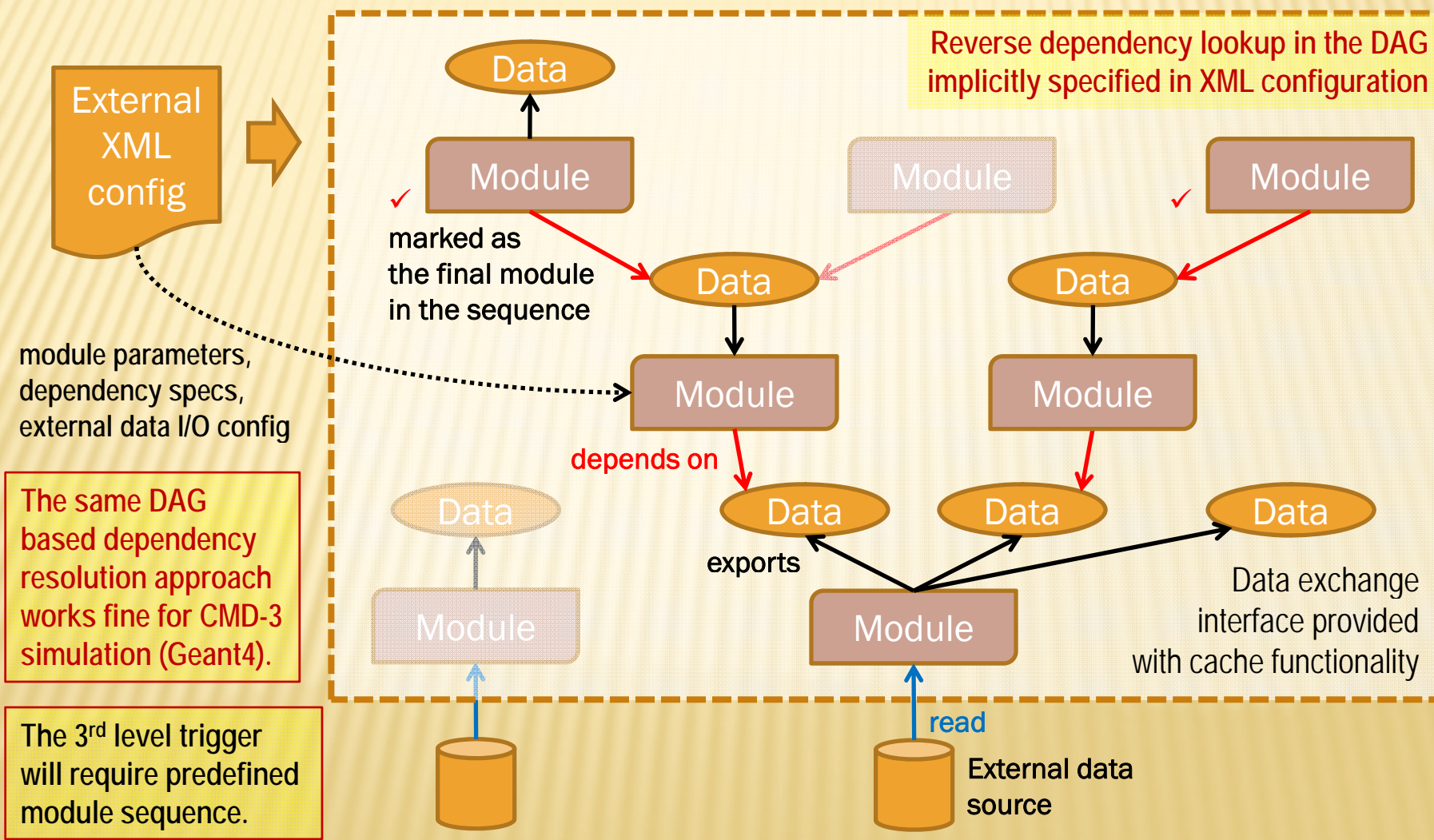
- The Framework provides the following tools for data access management exposed to the end users and module developers:
 - multilayer cache for automatic optimization of the modules call sequence in the reverse dependency lookup mode
 - It particular the cache can tuned to take into account the periodic structures within the input data (e.g., runs)
 - So called “proxy dictionary” mechanism associating a particular data instance in memory with a human readable name



SOFTWARE INTEGRATION PLATFORM

Cmd3Fwk Software Integration Framework

Development (DSO based) environment





Plans for the Next Releases

- The following items which are on the project TODO list were shifted to the next releases since they are not critical for testing the product in the development environment:
 - Fully functional support of static builds with predefined set of modules and/or XML configuration for the long term use in production
 - GUI tool for editing XML configuration with RelaxNG/XML Schema and interactive dependency tree visualization support
 - Standalone interactive tool for reviewing and filtering logger messages
 - Python API



Summary

- The project is now approaching the 1st publicly available production release v1.0 which will be provided with extensive set of technical documentation and the set of application building examples
- The functionality implemented in the h-dp-fwk up to now is enough to build a software integration solution for the offline data processing of a small or moderate scale HEP detector experiment
- Release v1.0 is planned to be thoroughly tested and validated within the framework of CMD-3 detector collaboration (BINP, Novosibirsk)
- An updated TODO list for the next release v1.1 should be commissioned shortly afterwards
- In addition the possibility of developing the sub-frameworks for the target fields outside the scope of HEP data processing will be investigated later this year
- For more details please refer to the documentation section of the official website: <http://hdpfwk.org/?docs>