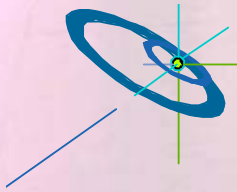
The Large Knowledge Collider (LarKC)
WP5 The Collider Platform

Georgina Gallizo (gallizo@hlrs.de)
Rolf Rabenseifner (rabenseifner@hlrs.de)
Lutz Schubert (schubert@hlrs.de)
Axel Tenschert (tenschert@hlrs.de)
http://www.hlrs.de

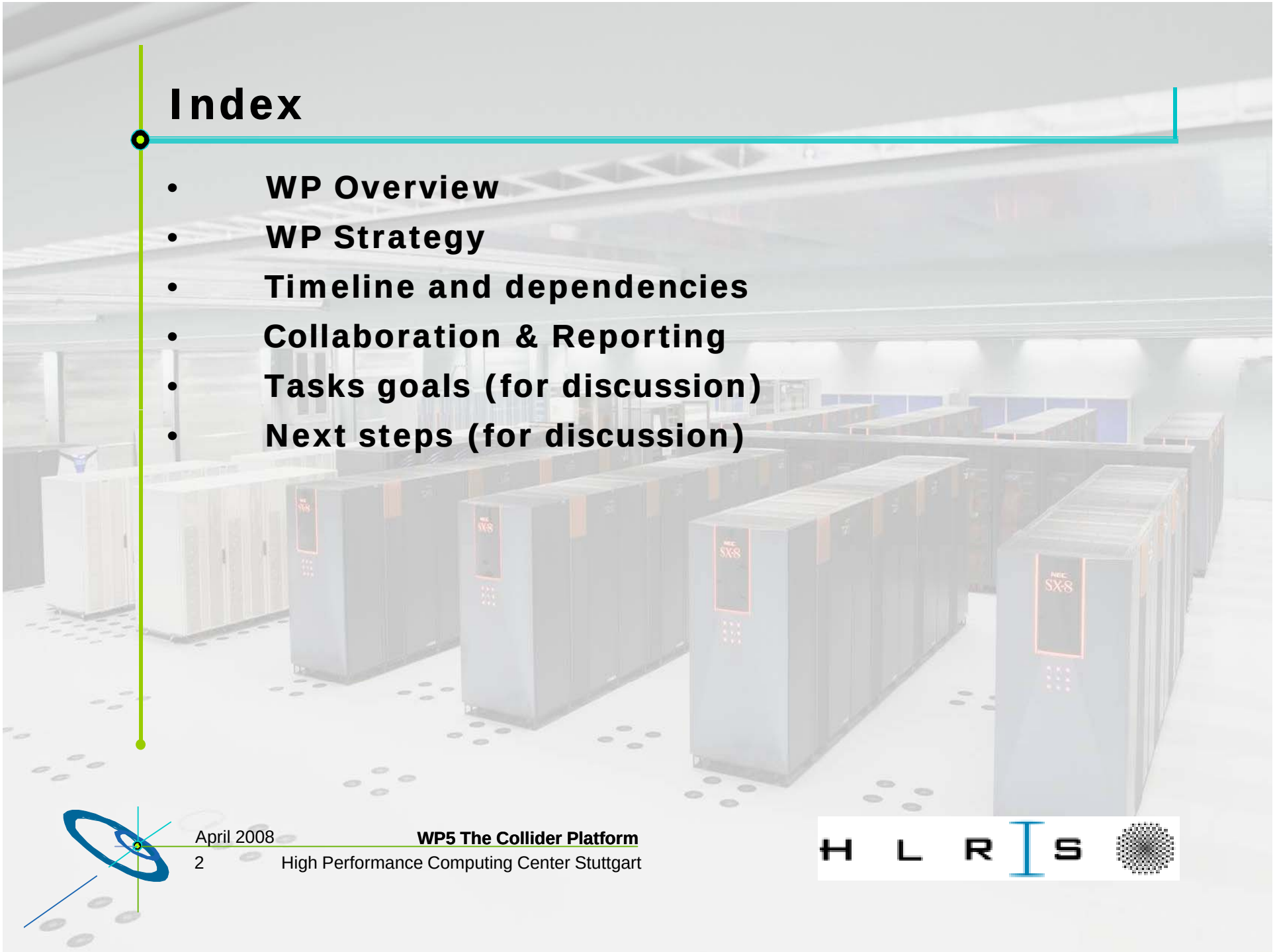
17 April 2008, LarKC Kick Off Meeting



HLRS

Index

- WP Overview
- WP Strategy
- Timeline and dependencies
- Collaboration & Reporting
- Tasks goals (for discussion)
- Next steps (for discussion)



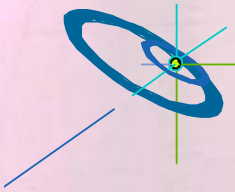
Overview

Work package number	WP5	**Start date or starting event**	M1
Work package title	**The Collider Platform**		
Activity type	RTD		
Participant id	CycEur	**HLRS**	Onto
Person-months per beneficiary	34	**57**	34

Objectives

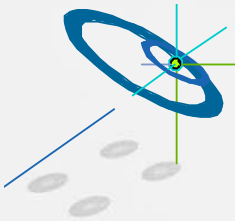
Develop an *open source, modular and distributed* platform for inference that will enable design, testing and exploitation of new reasoning techniques to be done at vastly reduced costs. A plug in architecture will support cooperation between heterogeneous, cooperating modules from partners and outside developers

Strategy



Overall Strategy

- **Target: flexible, extendable and scalable framework**
- **Some technologies (according to DoW):**
  - **Map/reduce** and similar paradigms, which foster the usage of software design patterns suitable for distributed/parallel computing
  - **Cluster** technologies, which do not require expensive connectivity between the nodes
  - **RDF** as a data-model for publication and interlinking of structured data from multiple heterogeneous data-sources
  - **Tuple spaces** computing, as a paradigm for communication and data sharing
  - **BOINC** and similar efforts for “computing at home” (as e.g. SETI@home and folding@home)



Tasks

- Task 5.1 State-of-the-art Survey, HLRS → D5.1
- Task 5.2 Early Release Prototype, CycEur → D5.2.1
- Task 5.3 Platform Architecture and Design, HLRS → D5.3.1, D5.3.2, D5.3.3
- Task 5.4 Platform realization, HLRS → D5.4.1, D5.4.2, D5.4.3
- Task 5.5 Platform validation, Onto → D5.5.1, D5.5.2, D5.5.3, D5.5.4
- Task 5.6 Development Infrastructure, HLRS → D5.6.1

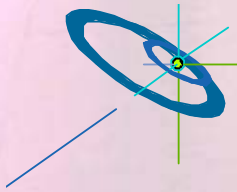
Deliverables (I)

Deliverable	Lead	PMs	Nature, Dis.level	Due date
D5.1 Summary of parallelization and control approaches…	HLRS	10	R, PU	M6
D5.2.1 Rapid prototype of the LarKC	CycEur	12	P, PU	M10
D5.3.1 Req. Analysis and report on lessons learned during prototype	CycEur	6	R, PU	M12
D5.3.2 Overall LarKC architecture and design v1	HLRS	18	R, PU	M18
D5.3.3 Final LarKC architecture and design	HLRS	12	R, PU	M36
D5.4.1 Initial release of the LarKC platform	HLRS	18	P, PU	M24
D5.4.2 Second release of the LarKC platform	HLRS	12	P, PU	M33
D5.4.3 Final release of the LarKC platform	HLRS	12	P, PU	M42

Deliverables (II)

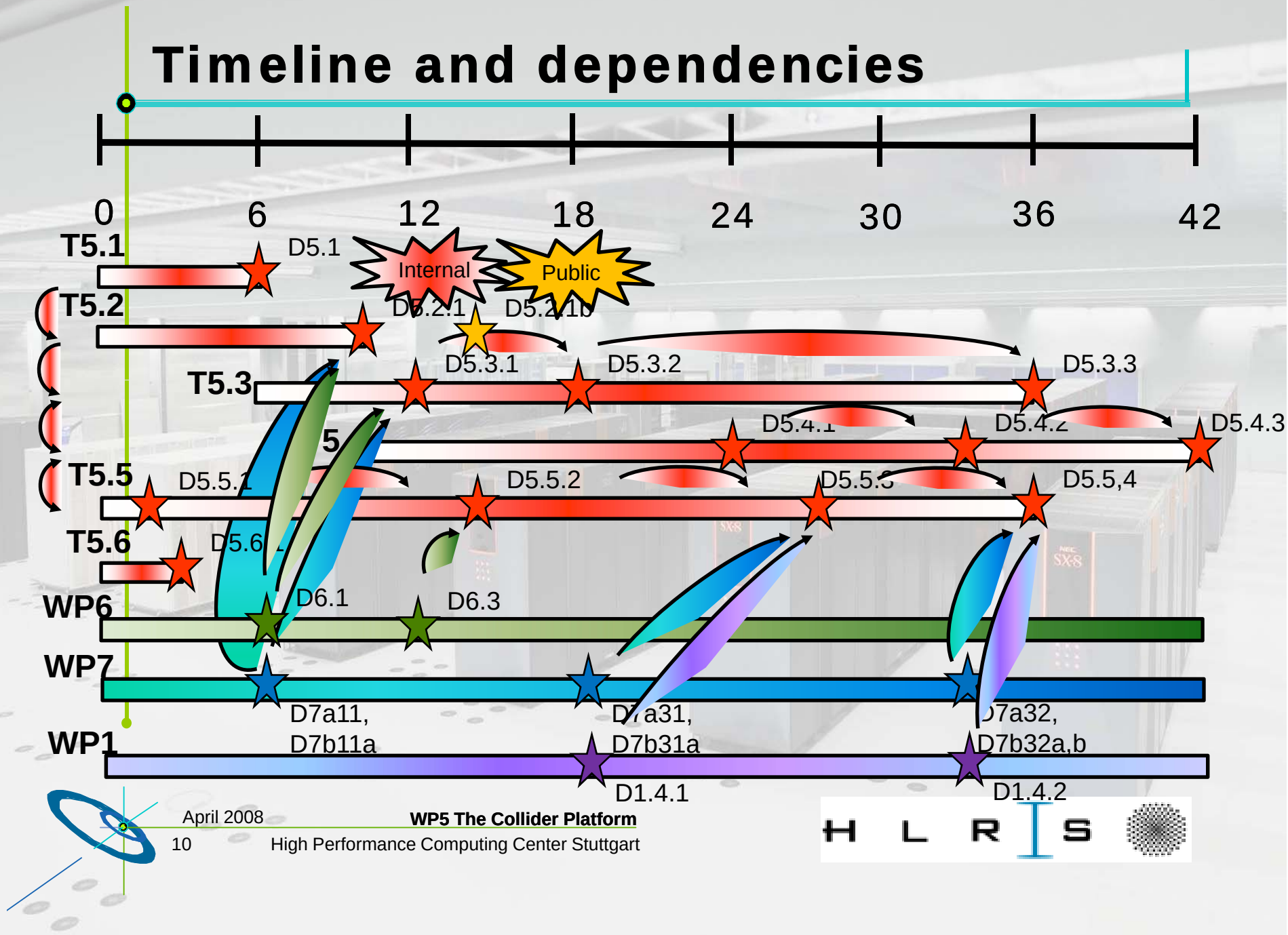
Deliverable	Lead	PMs	Nature, Dis.level	Due date
D5.5.1 Definition of validation goals for the prototyping phase	Onto	3	R, PU	M2
D5.5.2 Validation goals and metrics for the LarKC platform	Onto	6	P, PU	M14
D5.5.3 Report on platform validation and recommenndation for next version	Onto	9	R, PU	M28
D5.5.4 Report on platform validation and recommendation for final version	Onto	6	R, PU	M36
D5.6.1 LarKC development environment available	HLRS	1	O, PU	M3

Timeline and dependencies

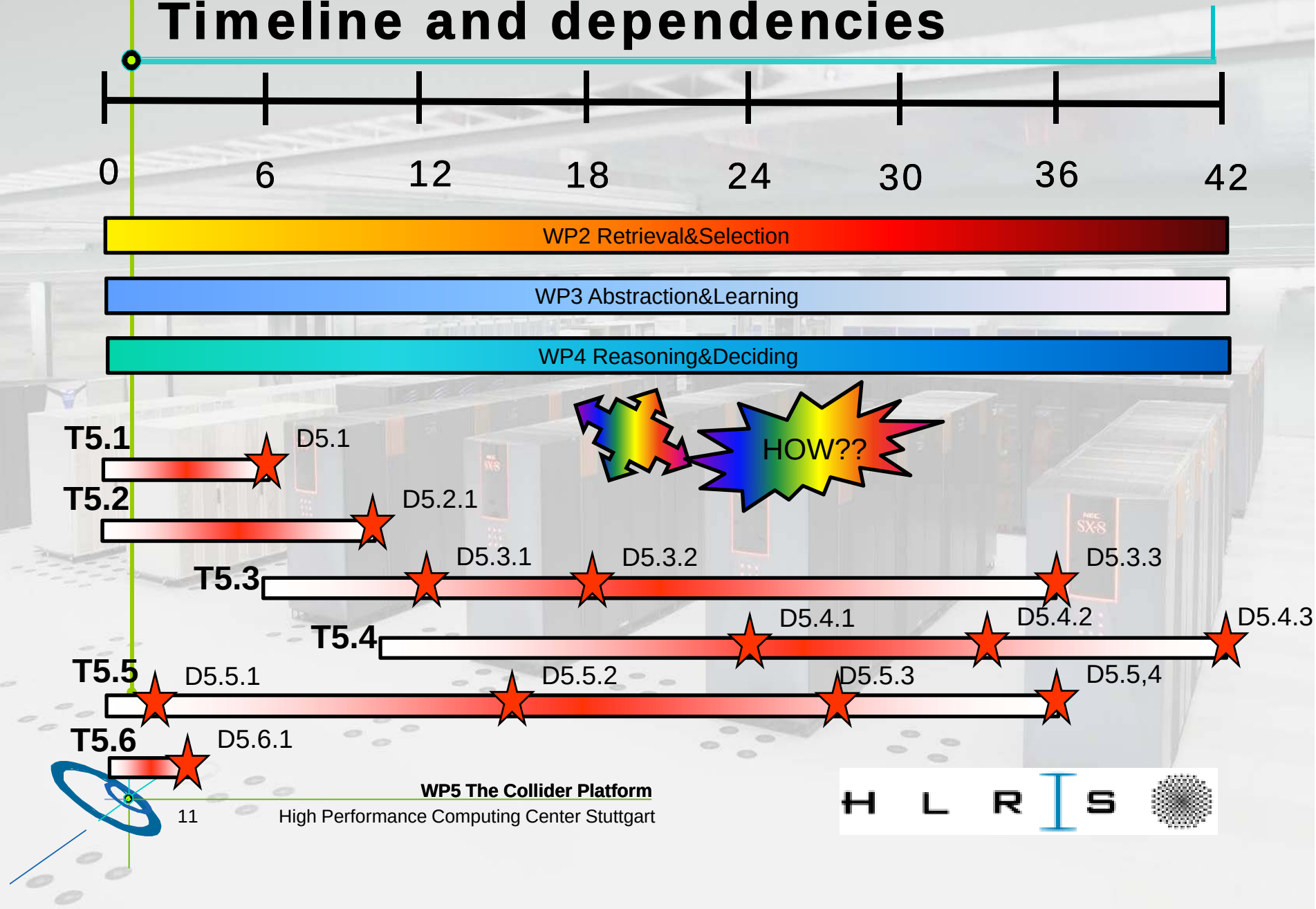


HLRS

Timeline and dependencies



Timeline and dependencies



Collaboration & Reporting

Collaboration & Reporting

Collaboration

- **WP5 mailing list**
- **Periodic Conference Calls**
  - **Intra-WP:** Coordinate the work of the partners involved in the WP
    - Every 2/3 weeks?
  - **Cross-WP:** Cross-WP communication to align strategies and coordinate feedback
    - Identify input needed: what and when => set conference calls as needed

Reporting

- **Partners reporting to WPLeader**
  - Align reporting procedure with Project Management procedure (Monthly Reports,...)
- **WPLeader reporting to Project Coordinator**
  - Align reporting procedure with Project Management procedure

Tasks goals

What do we want to achieve? What should be available at the end of each task? (Open discussion)

What do we want to achieve? (open discussion)

- **Task 5.1 State-of-the-art Survey, HLRS**
  - Analyze existing algorithms and applications for potential parallelization approaches and programming models
  - Focus on the new domain of the Large Knowledge Collider, on incomplete, stochastic inference => novel, previously unexplored forms of cooperation and parallelization

D5.1 Summary of parallelization and control approaches…	HLRS	10	R, PU	M6

- Participants
- Roles
- Goals
- …

What do we want to achieve? (open discussion)

- **Task 5.2 Early Release Prototype, CycEur**
  - Distributed reasoning platform based on the ResearchCyc inference engine
  - Focus on simplifying the modular, plug-in architecture for easier experimentation, removing Cyc-specific code not needed, and improving on the current limited support for parallel operation

D5.2.1 Rapid prototype of the LarKC	CycEur	12	P, PU	M10

- Internal release: M10
- Public release: M12-14
- Requires:
  - API design for:
    - Plug ins
    - Storage components
  - Plug ins (not yet from WP2,3,4)
  - Data (not yet from WP6,7)
- No parallelization is considered for this prototype:
  - In plug ins
  - In the platform?

What do we want to achieve? (open discussion)

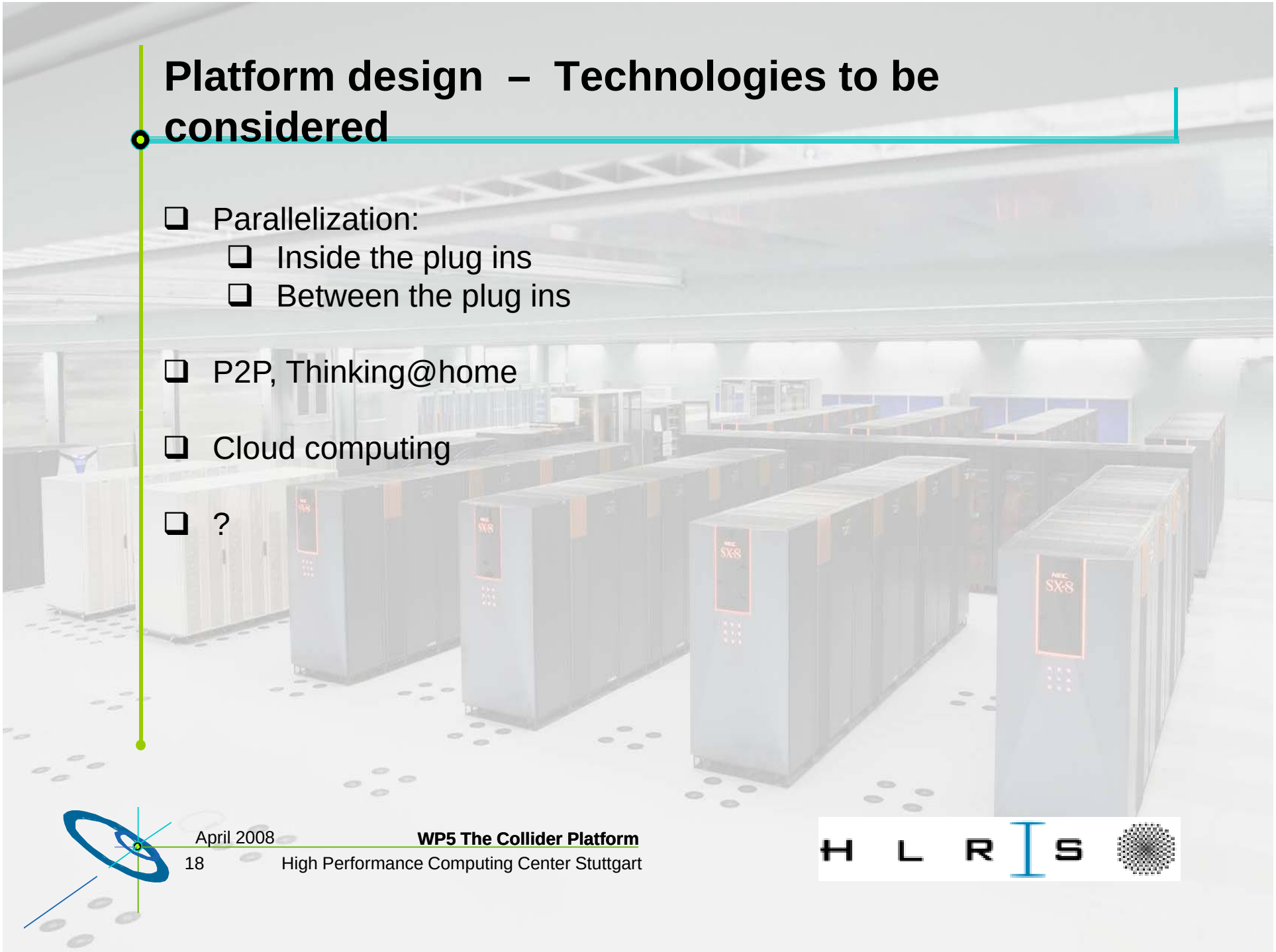
- **Task 5.3 Platform Architecture and Design, HLRS**
  - Iterate design of the Large Knowledge Collider.
  - Input from: use-case WPs (WP6, WP7), Task 5.1 (state-of-the-art survey) and Task 5.2 (early release prototype)
  - Interaction between designers and implementers of the platform and designers and implementers of the plug-ins
  - Address issues such as interface definitions for the all plug-in components of the overall algorithmic schema, and the division of computation between fine-grained, cluster based and wide-area distribution

D5.3.1 Req. Analysis and report on lessons learned during prototype	CycEur	6	R, PU	M12
D5.3.2 Overall LarKC architecture and design v1	HLRS	18	R, PU	M18
D5.3.3 Final LarKC architecture and design	HLRS	12	R, PU	M36

- Participants
- Roles
- Goals
- ...

Platform design – Technologies to be considered

- Parallelization:
  - Inside the plug ins
  - Between the plug ins
- P2P, Thinking@home
- Cloud computing
- ?

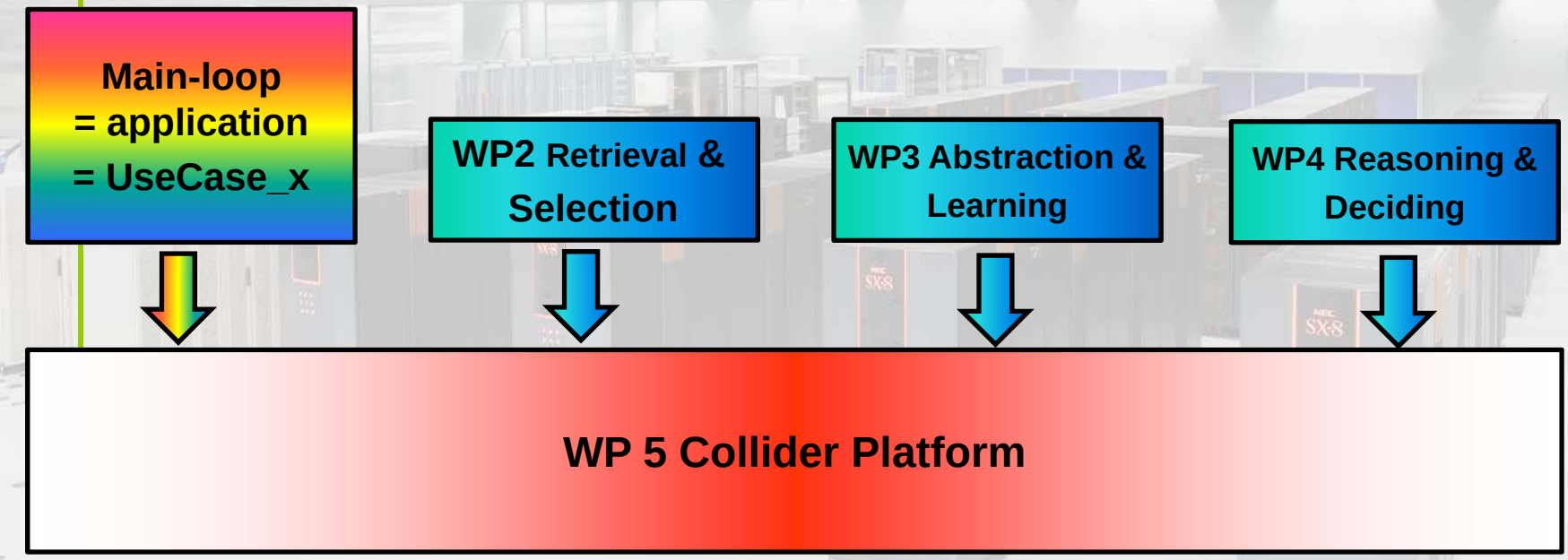


Platform design – Lessons learned

- Lessons learned from high-performance numerical simulations
  - In/out data only as vector of objects
    e.g., tan(n, in_angle_vector, out_tangens_vector)
    and **not** tan(in_angle, out_tangens)
- Never use linked lists
  - Use only arrays
  - Linked lists can not be parallelized
  - Nor optimized by the compiler
- Long loops through arrays of data are optimal
- Prefer several arrays of elements instead of array of structure
  - Reason: small memory bandwidth, cache lines “have no holes”
- Parallel programs are looking like a serial program,
  but all program steps are computed in parallel in a collective way
  - Application routines are **collective routines**
- Owner of 3rd party software deliver only executables,
  - never sources
  - and do not like to deliver .o files

Platform design – Major questions

- How can we plug-in the PLUG-INs and the MAIN-LOOP?
- How can we exchange data between main-loop and (between) plug-ins?
- How can we simply extend to parallel execution?



Platform design – Plug-in technology

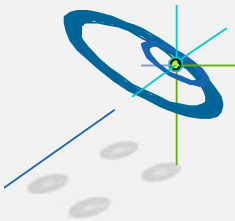
- How can we plug-in the PLUG-INs and the MAIN-LOOP?

Alternatives:

- **Use subroutines**
  - Plug-ins must be complied with proper platform.h file
  - .o file is used used for plugging it in.

  Advantages:
  - Maximal efficiency
  - Best basis for parallelization

  Disadvantages:
  - Drawbacks for 3rd party software
  - platform.h must not be modified (extensions are allowed)



Platform design – Plug-in technology

- How can we plug-in the PLUG-INs and the MAIN-LOOP?

Alternatives:

- **Web Services (statefull? stateless?)**
  - Process is spawned
  - Implementable with a special subroutine in the “subroutine concept”
  - i.e., both, “subroutine concept” and “web service” can co-exist

Advantages:

- Optimal for plugging in 3rd party modules
- The main loop can use it several times without restarting?

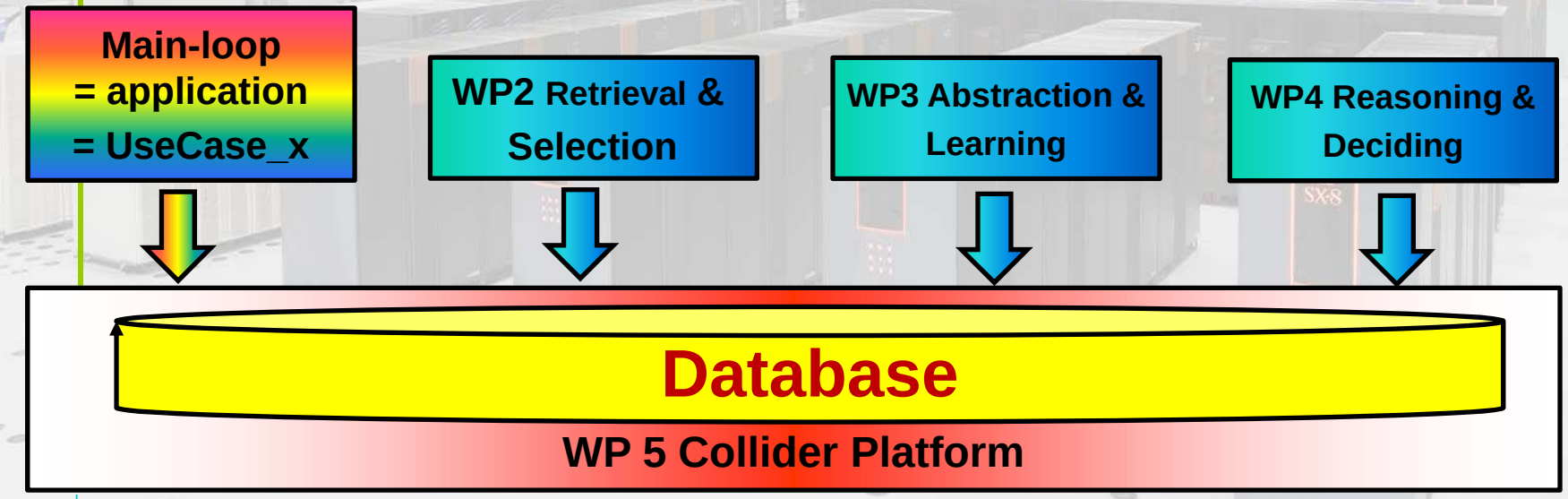
Disadvantages:

- Bad performance in startup and data-exchange
- Only prepared for parallelization inside of the module, not for efficient data-exchange between parallel processes of several plug-ins

Platform design – Major questions

- How can we plug-in the PLUG-INs and the MAIN-LOOP?
- How can we exchange data between main-loop and (between) plug-ins?

- Is the database part of the Collider Platform?
- Vector interfaces!
- Vector of RDF, of OWLs, of xxx, of yyy, of zzz, …
  or simply vector of generalized n-tuples



Platform design – Major questions

- How can we plug-in the PLUG-INs and the MAIN-LOOP?
- How can we exchange data between main-loop and (between) plug-ins?

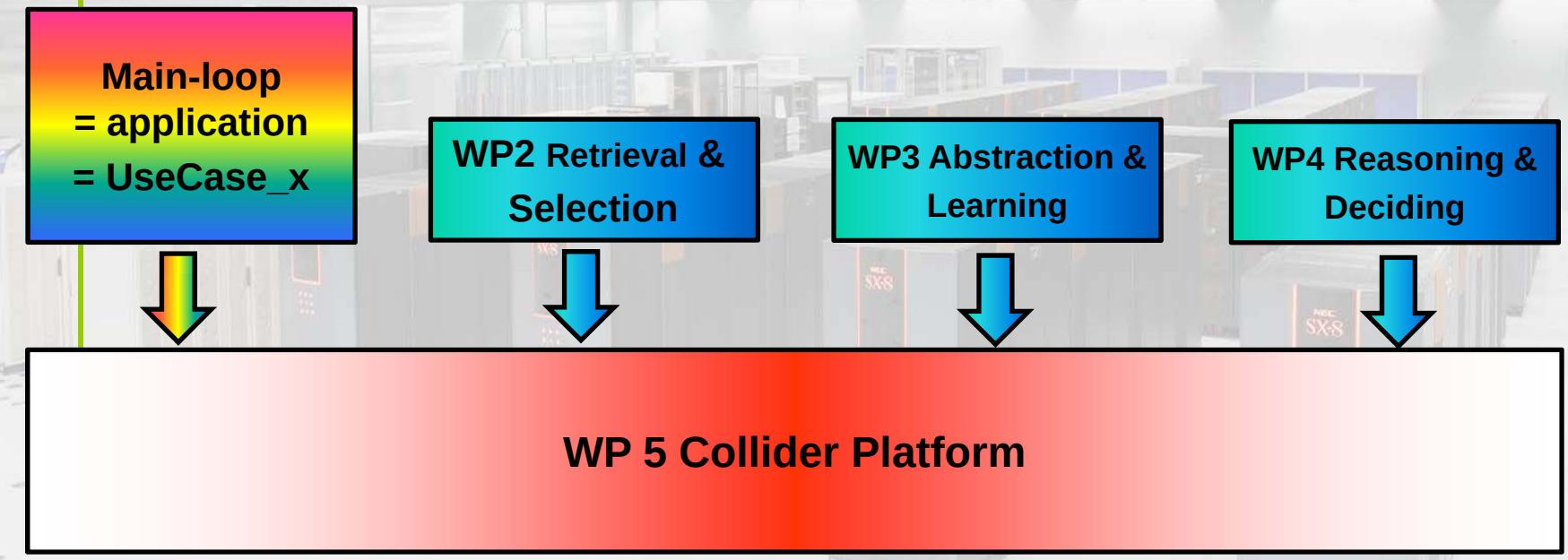
- Is the database part of the Collider Platform?
- Vector interfaces!
- Vector of RDF, of OWLs, of xxx, of yyy, of zzz, …
  or simply vector of generalized n-tuples

- We should have efficient interfaces for the case that, e.g.
  - All predicates are identical (e.g., ***isCityInState***) for the whole vector
  - Or entry j in the n-tuples are identical for the whole vector

- We should have additional interfaces for (meta) data like
  - History of the data, references, …, see talk of …

Platform design – Major questions

- How can we plug-in the PLUG-INs and the MAIN-LOOP?
- How can we exchange data between main-loop and (between) plug-ins?
- How can we simply extend to parallel execution?



What do we want to achieve? (open discussion)

- **Task 5.4 Platform realization, HLRS**
  - Release versions of LarKC
  - Input from will Task5.2 and the architecture from Task5.3
  - Three major foci:
    1. Refining the control architecture to improve inference control flow
    2. Adapting the distributed middleware for more effective, efficient interoperation of operating modules
    3. Advancing the range and ease of implementation of tightly coupled algorithms by exploiting techniques for fine-grained parallel and vector systems

D5.4.1 Initial release of the LarKC platform	HLRS	18	P, PU	M24
D5.4.2 Second release of the LarKC platform	HLRS	12	P, PU	M33
D5.4.3 Final release of the LarKC platform	HLRS	12	P, PU	M42

- Participants
- Roles
- Goals
- ...

What do we want to achieve? (open discussion)

- **Task 5.5 Platform validation, Onto**
  - Investigation of performance curves as a function of available resources and problem scale
  - Evaluate performance and bottlenecks as observed in the case studies (WP6, WP7)
  - Experiment with the forms of inference in well-studied cognitive architectures (e.g. ACT-R, CogAff, Society of Mind) to gauge the benefits of the Large Knowledge Collider
  - Alignment with Task 1.4, which develops an overall evaluation framework suitable for the application settings in which LarKC technology will be used

What is measured is what matters (Dr. Mark Greaves, Vulcan)

D5.5.1 Definition of validation goals for the prototyping phase	Onto	3	R, PU	M2
D5.5.2 Validation goals and metrics for the LarKC platform	Onto	6	P, PU	M14
D5.5.3 Report on platform validation and recommenndation for next version	Onto	9	R, PU	M28
D5.5.4 Report on platform validation and recommendation for final version	Onto	6	R, PU	M36

What do we want to achieve? (open discussion)

- **Task 5.6 Development Infrastructure, HLRS**
  - Set-up and an internal communication tool (configuration mgmt, developer exchange, release mgmt)
  - Open it to the public: report bugs, make feature requests and other comments

  => allow to efficiently develop and quickly fix bugs and realize the infrastructure to ensure good responsiveness on above requests in a reasonable time11

D5.6.1 LarKC development environment available	**HLRS**	**1**	**O, PU**	**M3**

- Participants
- Roles
- Goals
- …

Next steps

Next steps and more inmediate action points, responsible partner,...

Next steps (for discussion)

- **D5.5.1 Definition of validation goals for the prototyping phase (Onto) (M2, End May 08)**
  - Nature: Report; Diss. level: Public
  - Obj: Document steering the prototyping phase in an efficient way to make sure the input delivered to the architecture activity is relevant
  - Participants: Onto, ?
  - Planning:
    - …
    - Final version for internal review: 02/05/2008?
    - Final version for submission to EC: 30/05/2008
  - Identification of risks:
    - Too early deadline?
    - Dependencies with T1.4?: The validation will take into account the results achieved in T1.4, in which a novel evaluation procedure for the Web-compliant methods developed in LarKC will be generated. D1.4.1 – M18
    - …
    - TO BE DISCUSSED
- **Next steps and more inmediate action points, responsible partner,...**

Next steps (for discussion)

- **D5.6.1 LarKC Development Environment Avilable (Typo in page 65, name of deliv.) (M3, End June 08)**
  - Nature: Other; Diss. level: Public
  - Obj: Configured project within the existing GFORGE environment at HLRS
  - Participants: HLRS
  - Planning:
    - …
    - No need for internal review
    - Final version: 30/06/2008
  - Identification of risks:
    - …
    - TO BE DISCUSSED

- **Next steps and more inmediate action points, responsible partner,...**

Next steps (for discussion)

- **D5.1 Summary of parallelization and control approaches and their exemplary application for selected algorithms or applications (HLRS) (M6, End Sep. 08)**
  - Nature: Report; Diss. level: Public
  - Obj: Summary of parallelisation and control approaches and their exemplary application for selected algorithms or applications
  - Participants: HLRS, input needed from other partners
  - Planning:
    - …
    - Final version for internal review: 01/09/2008?
    - Final version for submission to EC: 30/09/2008
  - Identification of risks:
    - …
    - TO BE DISCUSSED

- **Next steps and more inmediate action points, responsible partner,...**

Next steps (for discussion)

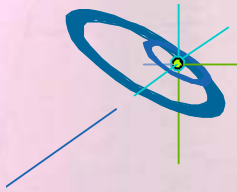
- **D5.2.1 Rapid prototype of the LarKC (CycEur) (M10, End Jan. 09)**
  - Nature: Prototype; Diss. level: Public
  - Obj: Initial prototype based on existing background software from involved partners to guide architecture process, reduce risk and show feasibility. The functionality of this prototype will already be informed by the requirements formulated in the use-cases
  - Participants: CycEur, HLRS,Onto
  - Planning:
    - …
    - Need for internal review or testing (prototype)?
    - Final version for submission to EC: 30/01/2009
  - Identification of risks:
    - Dependencies with D5.1, due M6?
    - Dependencies with D6.1, D7a1.1 and D7b.1.1a, use cases requirements, due M6?
    - …
    - TO BE DISCUSSED

- **Next steps and more inmediate action points, responsible partner,...**

Thanks

Georgina Gallizo (gallizo@hlrs.de)
Rolf Rabenseifner (rabenseifner@hlrs.de)
Lutz Schubert (schubert@hlrs.de)
Axel Tenschert (tenschert@hlrs.de)
http://www.hlrs.de

17 April 2008, LarKC Kick Off Meeting



HLRS