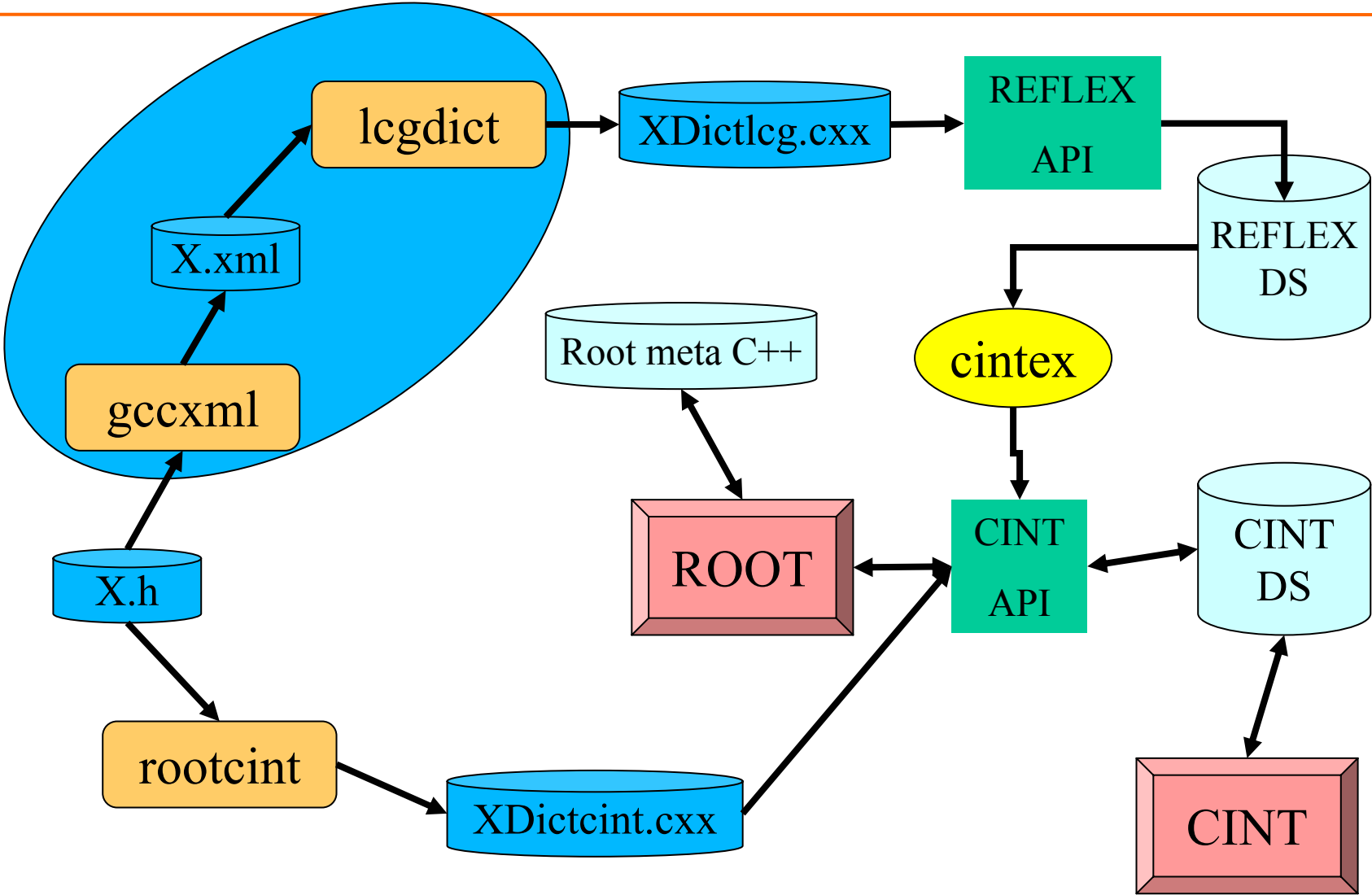


**DICTINT**  
**Dictionary, Reflex,CINT,Python**

---

**Philippe Canal**  
**Masa Goto**  
**Stefan Roiser**  
**Wim Lavrijsen**

# Dictionaries : situation today

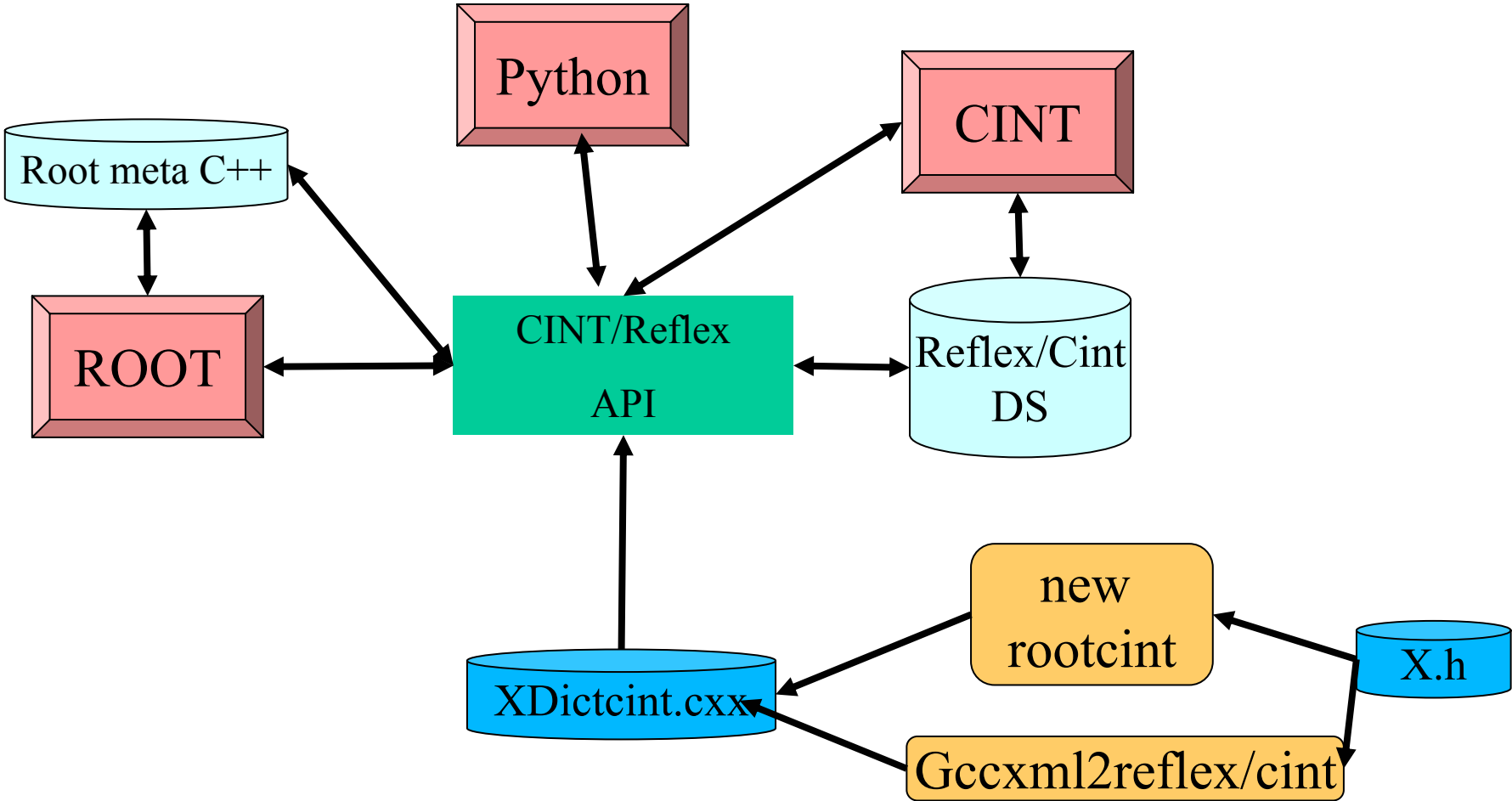


# CINT/Reflex workshop

---

- A very important workshop has been scheduled for May 2->7 to discuss the integration of Reflex and CINT.
  - Fons, Markus, Masa, Philippe, Rene, Stefan
- We hope to converge on a C++ DS taking advantage of Reflex and the current redesign of CINT by Masa.
- If successful, Cintex will not be required anymore.

# Dictionaries : situation in the future



# CINT & Reflex

	A	B	Formula Bar	C
1	<b>Description</b>	<b>CINT</b>		<b>Reflex</b>
2	Type (class, enum, struct, union)	char G__tagtable::type[]		TYPE TypeBase::typeType
3	Name of type	char *G__tagtable::name[]		const char * TypeName::m_name
4	Hash value of Type	int G__tagtable::hash[]		TypeName * Type::m_typeName
5	Size of Type	int G__tagtable::size[]		size_t TypeBase::m_size
6	DataMembers	struct G__var_array *G__tagtable::memvar[]		vector<Member> ScopeBase::m_dataMembers
7	FunctionMembers	struct G__ifunc_table *G__tagtable::memfunc[]		vector<Member> ScopeBase::m_functionMembers
8	InheritanceTree	struct G__inheritance *G__tagtable::baseclass[]		vector<Base> Class::m_bases
9	offset to vtable	int G__tagtable::virtual_offset[]		
10	??	G_SIGNEDCHAR_T G__tagtable::globalcomp[]		indicates the type of i
11	??	G_SIGNEDCHAR_T G__tagtable::iscpplink[]		indicates the type of i
12	class is abstract	char G__tagtable::isabstract[]		unsigned int Class::m_modifiers
13	??	char G__tagtable::protectedaccess[]		indicates wether the c
14	line of code (start of type)	int G__tagtable::line_number[]		PropertyList TypeBase::m_propertyList
15	file (where type is declared)	short G__tagtable::filenum[]		PropertyList TypeBase::m_propertyList
16	declaring scope	short G__tagtable::parent_tagnum[]		Scope ScopeBase::m_declaringScope
17	??	unsigned char G__tagtable::funcs[]		meta infor about cons
18	typedef for type exists	char G__tagtable::istypedefed[]		Type Typedef::m_typedefType (??)
19	??	char G__tagtable::istrace[]		whether the class sho
20	??	char G__tagtable::isbreak[]		whether the CINT deb
21	number of all tags defined	int G__tagtable::alltag		hash_map<const char*,TypeName*> TypeName::sTypes().size()
22	tags of friends	struct G__friendtag *G__tagtable::friendtag[]		
23	c++ comment	struct G__coment_info G__tagtable::comment[]		PropertyList TypeBase::m_propertyList
24	??	G__incsetup G__tagtable::incsetup_memvar[]		reference to the Diction
25	??	G__incsetup G__tagtable::incsetup_memfunc[]		reference to the Diction
26	??	char G__tagtable::rootflag[]		Which ROOT flags was
27	??	struct G__rootspecial *G__tagtable::rootspecial[]		Store extra informatio
28	type has constructor defined	char G__tagtable::isctor[]		vector<Member> Class::m_constructors
29	??	int G__tagtable::defaulttypenum[]		point to a typedef wh
30	user parameters	void *G__tagtable::userparam[]		PropertyList TypeBase::m_propertyList
31	library name	char *G__tagtable::libname[]		PropertyList TypeBase::m_propertyList
32	pointer to member (offset)	long G__var_array::p[]		size_t DataMember::m_offset vector<Member>
33	number of members	int G__var_array::allvar		ScopeBase::m_dataMembers.size()

# CINT, Dictionaries & Interpreters

---

- Main Goal
  - Improve the conformity of the dictionary generator with the C++ standard without sacrificing performance.
- Move to CINT 6 (enhanced code executor)
- Understand the issues surrounding the incorporation of Reflex (and gcc\_xml)
  - Portability of gcc\_xml to all the CINT supported platform
  - Size and ease of installation of gcc\_xml distribution
  - Backward compatibility issues of the CINT API
  - Performance issues (as far as interpreter is concerned)
  - Structure of the collaboration with Masa (to CVS or not to CVS).
  - Update of the ROOT Meta classes to leverage Reflex API

# Meta package

---

- Improve the support for loading a library implementing a nested collection after the collection has already been emulated.
  - Introduce either a TClassRef or insure that TClass objects are never deleted
- Test (or decide to abandon the idea) changing ProcessLine to return a Long64\_t
- Check if the autoloader now works for classes in namespaces
- Add a TClass:SetCanSplit or equivalent
- Add support for virtual derivation

# CINT and Dictionaries

---

- Interpreted vs. Compiled function
  - Resolve issues of asymmetry in their usage
- `#include <stdio.h>` vs. CINT's embedded functions
- Improve support for typedef to function type
- Improve support for using statement when used for class and function templates.
- Add proper support for private and protected inheritance
- Resolve issues with non-qualified name
  - `list<std::list<B> >` vs `list<list<B> >`
- Investigate a few hard to reproduce core dumps.