



caBIOperl: a new Perl API to the NCI's Biomedical Domain Object Middleware

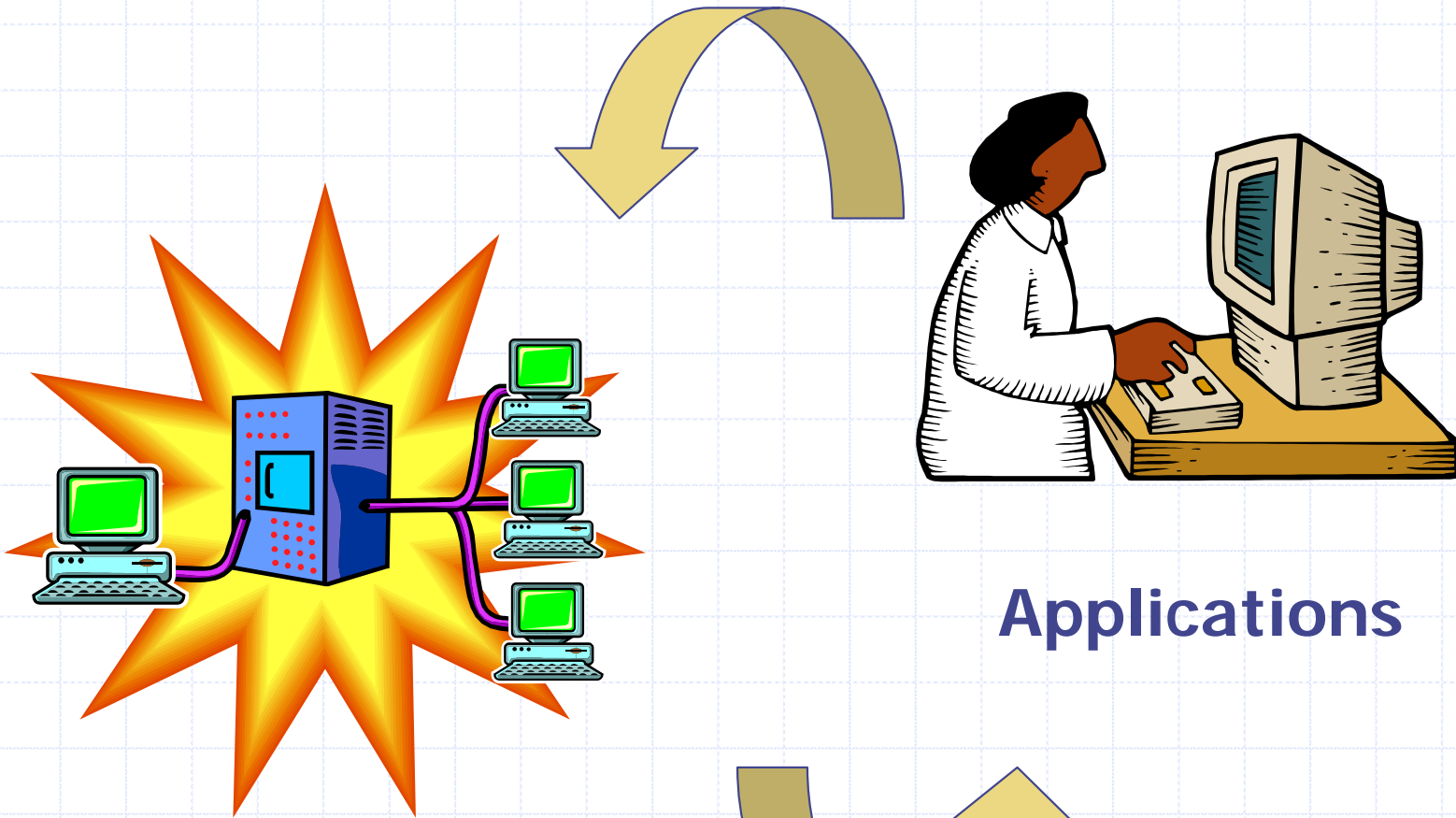


Glasgow, UK

Shan Jiang
Ying Long
Gene Levinson
Peter A. Covitz



NCICB organization



Core Infrastructure

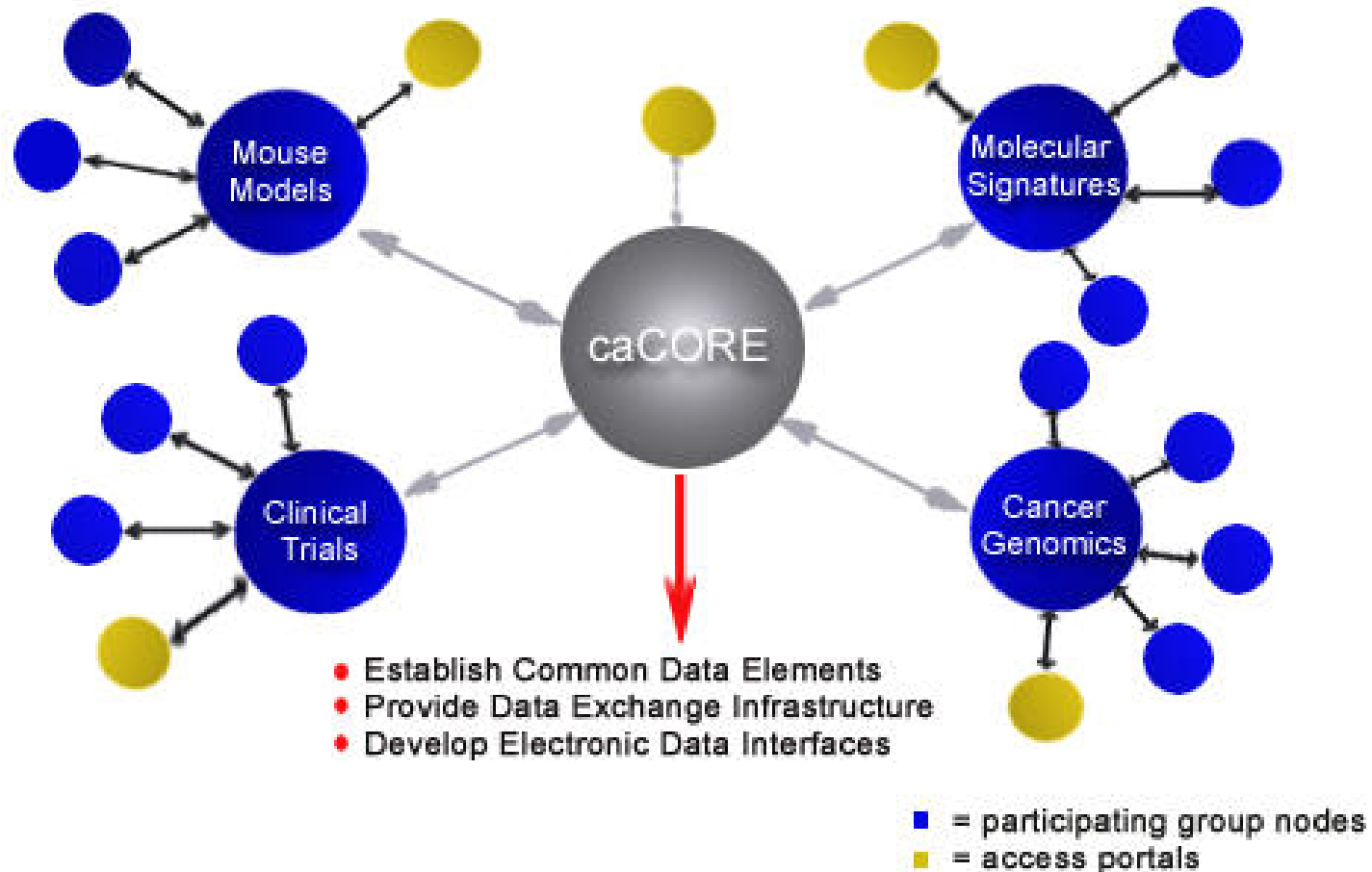
•caCORE

Applications

caCORE plays a central role...

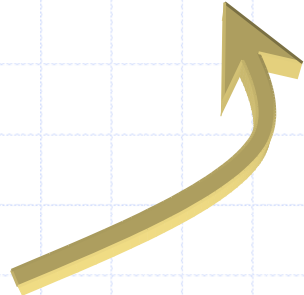
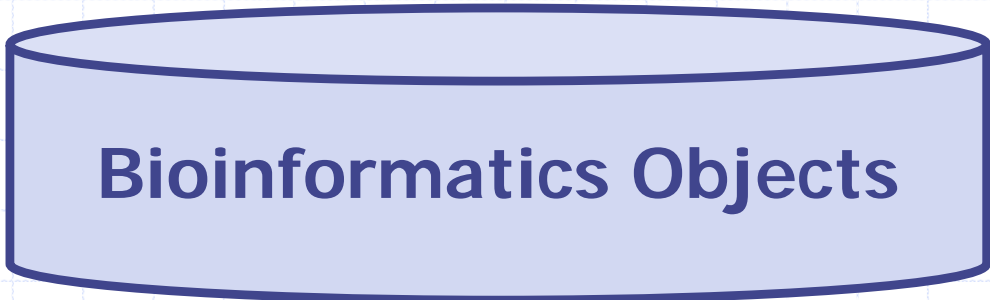
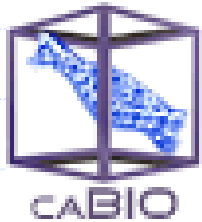
NCI Center for Bioinformatics

Building common tools, common architecture and common standards

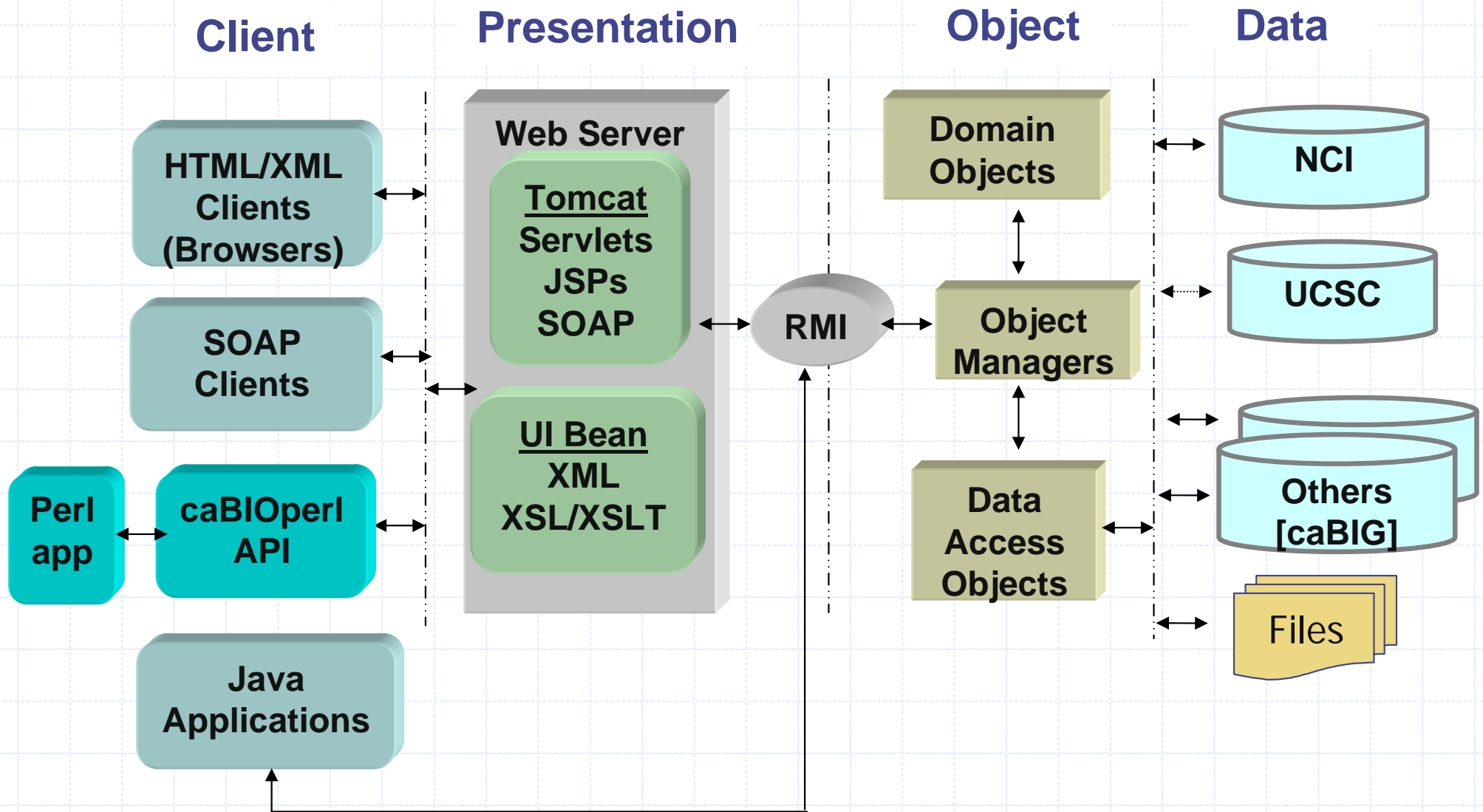


caCORE

Scientific Applications



caBIO architecture



caBIO APIs

◆ Java

- Use caBIO domain objects within your Java program
- Data automatically retrieved from caBIO servers

◆ SOAP-XML

- SOAP client in any language/environment can send request to NCICB server for object data
- SOAP-XML envelope and payload returned

◆ HTTP-XML

- Properly formed URLs in any web browser/client can retrieve XML-formatted object data directly

◆ caBIOperl

- Latest addition: Perl wrapper of a SOAP interface

What is caBIOperl?

- ◆ Essentially a Perl version of the caBIO Java API
- ◆ Instantiate caBIO data objects directly as Perl objects, just as if working with caBIO in Java
- ◆ Under the hood caBIOperl uses caBIO SOAP-XML interface: parses all the XML and populates objects with data for you
- ◆ Developed mostly by Shan Jiang, SAIC employee at the NCI Center for Bioinformatics

What can caBIOperl do for you?

- ◆ Presents an object oriented interface to access caBIO objects in Perl; frees Perl programmers from manipulating SOAP and XML
- ◆ Query broad range of data sources by calling methods, without SQL
- ◆ Navigate between objects using Perl code

Objects and Documentation

[Agent](#)
[Anomaly](#)
[Chromosome](#)
[ClinicalTrialProtocol](#)
[Clone](#)
[CMAPOntology](#)
[CMAPOntologyRelationship](#)
[Disease](#)
[ExpressionFeature](#)
[ExpressionMeasurement](#)
[ExpressionMeasurementArray](#)
[Gene](#)
[GeneAlias](#)
[GeneHomolog](#)
[GoOntology](#)
[GoOntologyRelationship](#)
[Histopathology](#)
[Library](#)
[MapLocation](#)
[Organ](#)
[OrganRelationship](#)
[Pathway](#)
[Protein](#)
[ProteinHomolog](#)
[Protocol](#)
[ProtocolAssociation](#)
[Sequence](#)
[SNP](#)
[Target](#)
[Taxon](#)
[Tissue](#)
[Vocabulary](#)

- [Object creation and retrieval methods](#)
- [Attributes and related methods](#)
- [Associations and related methods](#)
- [EXPORT](#)
- [SEE ALSO](#)
- [AUTHOR](#)
- [COPYRIGHT AND LICENSE](#)

NAME

gov::nih::nci::caBIO::bean::Agent - Perl extension for Agent.

SYNOPSIS

```
use gov::nih::nci::caBIO::bean::Agent;  
use gov::nih::nci::caBIO::bean::AgentSearchCriteria;  
  
# create a AgentSearchCriteria instance to specify search elements  
my $search = new gov::nih::nci::caBIO::bean::AgentSearchCriteria;  
# specify a search element  
$search->setId(1080);  
# create a dummy bean instance to invoke its search methods  
my $bean = new gov::nih::nci::caBIO::bean::Agent;  
# return all beans that meets the search element  
my $beanlist = $bean->searchAgent($search);  
...
```

Example 1: Pathway Search

◆ For the gene symbol BRCA1:

- Find all pathways associated with the gene symbol BRCA1
- Retrieve a list of all available pathway descriptions as well as SVG filenames

Instantiate the search object(s):

```
@symbols = qw(BRCA1) ;  
$bean = new gov::nih::nci::caBIO::bean::Gene() ;  
  
foreach $sym (@symbols) {  
  
    $searchCriteria =  
    new gov::nih::nci::caBIO::bean::GeneSearchCriteria() ;  
  
    $searchCriteria->setName($sym) ;  
    $result = $bean->search($searchCriteria);  
}
```

Create an array for the results:

```
@genes = $result->getResultSet() ;
```

```
foreach $gene (@genes) {
```

Print pathway description and SVG filename:

```
@pathways = $gene->getPathways() ;  
if (@pathways) {  
  foreach $pw (@pathways) {  
    printf "  Description: %s\n",  
    $pw->getDisplayValue() ;  
    $pw->getPathwayDescription() ;  
    printf "  SVG: %s\n",  
    $pw->getName() ;  
  }  
}
```

Output from a sample script:

Retrieving pathways for gene 'BRCA1'...

Found caBIO Gene BRCA1

Found associated Taxon: Homo sapiens

Found 5 associated Pathway(s):

Description: ATM Signaling Pathway

SVG: h_atmPathway

Description: Role of BRCA1, BRCA2 and ATR in Cancer Susceptibility

SVG: h_atrbrcaPathway

Description: BRCA1-dependent Ub-ligase activity

SVG: h_bard1Pathway

Description: CARM1 and Regulation of the Estrogen Receptor

SVG: h_carm-erPathway

Description: Cell Cycle: G2/M Checkpoint

SVG: h_g2Pathway

Example 2: Get Multiple Attributes

◆ For the gene symbol ATM, get:

- Title, LocusLink, ClusterID
- GoOntologies
- SNPs
- Map Locations
- Proteins
- Gene Homologs
- Gene Aliases

Create instance and search criterion:

```
sub getGenes {
```

```
    my $bean = new gov::nih::nci::caBIO::bean::Gene;
```

```
    my $searchCriteria =
```

```
    new gov::nih::nci::caBIO::bean::GeneSearchCriteria;
```

```
    $searchCriteria->setName($symbol);
```

```
    $result = $bean->search($searchCriteria)
```


Process the result set:

```
my @rs = $result->getResultSet;
my $bean2;
foreach $bean2 (@rs) {
    print "Name = ".$bean2->getName."\n";
    print "Title = ".$bean2->getTitle."\n";
    print "LocusLinkId = ".$bean2->getLocusLinkId."\n";
    print "ClusterId = ".$bean2->getClusterId."\n\n";
}
```

Partial output from a sample script:

Name = ATM

**Title = ataxia telangiectasia mutated (includes
complementation groups A, C and D)**

LocusLinkId = 472

ClusterId = 526394

Name = Atm

**Title = ataxia telangiectasia mutated homolog
(human)**

LocusLinkId = 11920

ClusterId = 5088

Future versions...

◆ Next version:

- Nested queries
- Integrate with caMOD

◆ Further out:

- EVS
- caDSR
- Possibly others– please let us know!

Links:

◆ download:

<http://ncicb.nci.nih.gov/download>

◆ caBIO project page:

<http://ncicb.nci.nih.gov/core/cabio>

◆ NCICB application support:

voice: (1) 301-451-4384

email: ncicb@pop.nci.nih.gov

Acknowledgements



**Shan Jiang
Ying Long
Peter Covitz**

caBIO team, SAIC

Systems and support teams, Terrapin Systems

caBIO user and developer community!