Ian Mitchell, IBM Distinguished Engineer, CICS Transaction Server Architect.
Rob Nicholson, IBM Senior Technical Staff Member, MBCS, CEng, CITP.

Scripting in an Enterprise Application Environment

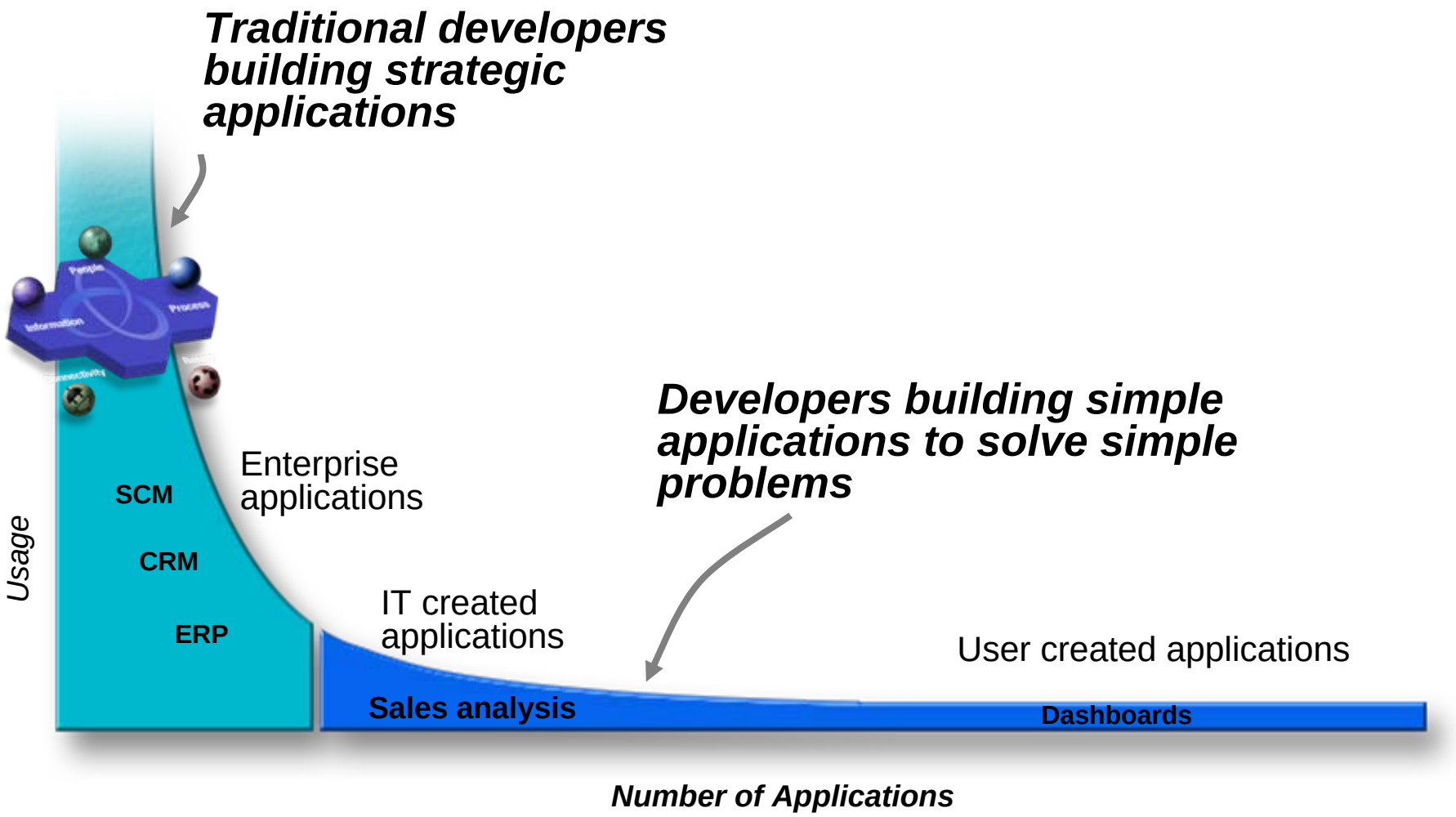
Agenda

- Why Scripting?
- Why PHP?
- What is PHP?
- Our PHP-on-Java implementation.
- PHP in CICS

Agenda

- Why Scripting?
- Why PHP?
- What is PHP?
- Our PHP-on-Java implementation.
- PHP in CICS

The Application Landscape



Requirements:

Quick to:

- Learn
  - Simple, forgiving language
  - Lots of examples, books
  - Google -> cut -> paste programming.
- Build version 1.
  - Powerful libraries, frameworks, patterns, idioms.
- Evolve and change.
  - No compile or deploy step.

Agenda

- Why Scripting?
- Why PHP?
- What is PHP?
- Our PHP-on-Java implementation.
- PHP in CICS

PHP, the language

- **Imperative**, **Procedural** and **Object Oriented**.
- **Dynamic**, **weakly typed** scripting language.
- Syntactically similar to C and Java.
- Server side web scripting and General Scripting.
  - Can be embedded in HTML.

```
<html> <head><title>PHP Example</title></head>
  <body>
    <?php $myVar=$_GET["myparm"]; ?>
    <p>Hello from HTML</p>
    <?php echo "<p>myparm=$myVar</p>" ?>
  </body>
</html>
```

PHP on the Public Internet

- **20M+ web domains use PHP (~ 1/3 of the internet)**
- **3M+ Programmers know PHP**
- **Significant web properties built on PHP**

TIOBE Programming Community Index (Oct 2009)

Position Oct 2009	Position Oct 2008	Delta in Position	Programming Language	Ratings Oct 2009	Delta Oct 2008	Status
1	1	=	Java	18.718%	-2.23%	A
2	2	=	C	16.891%	+1.33%	A
3	5	⇈	PHP	10.390%	+1.78%	A
4	3	↓	C++	9.911%	-1.04%	A
5	4	↓	(Visual) Basic	8.729%	-1.08%	A
6	8	⇈	C#	4.433%	+0.67%	A
7	6	↓	Python	3.914%	-0.65%	A
8	7	↓	Perl	3.776%	-0.64%	A
9	11	⇈	JavaScript	3.033%	+0.36%	A
10	10	=	Ruby	2.458%	-0.40%	A

PHP Community by the numbers

~ 150 programmers

(Source php.net svn)

19 Frameworks

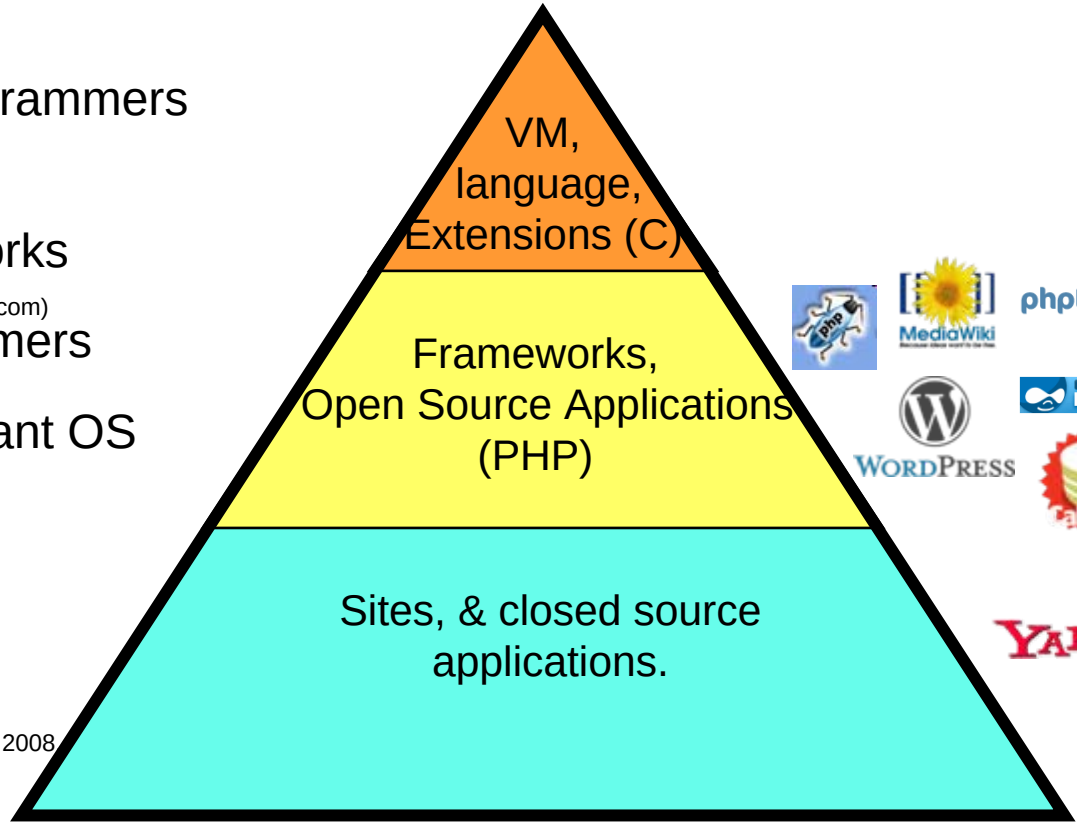
(source phpframeworks.com)

1000s of programmers

100s of Significant OS applications

3 to 5 Million Programmers

(source Gartner, 2007 Zend, 2008










Ning

Java Community is 7M programmers

(source Sun Nicrosystems 2008)

PHP inside the firewall?

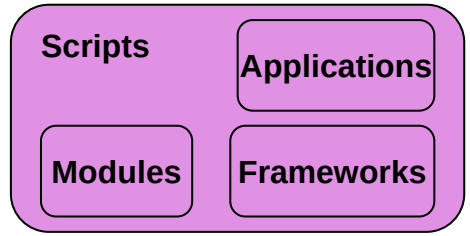
Gartner (Dec 2007)

- PHP Developers to grow from 3 to **5.5 million** by 2013
- PHP Developers in Commercial or Corporate IT to grow from 13% to **60%** by 2013
- "Pay special attention to opportunities to leverage PHP in combination with Java development efforts"

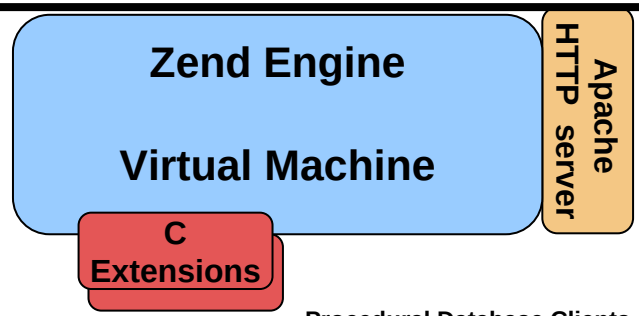
Agenda

- Why Scripting?
- Why PHP?
- What is PHP?
- Our PHP-on-Java implementation.
- PHP in CICS

PHP Reference Implementation



C Language



Vast array of modules implemented in PHP.

Rich set of extensions implemented in C

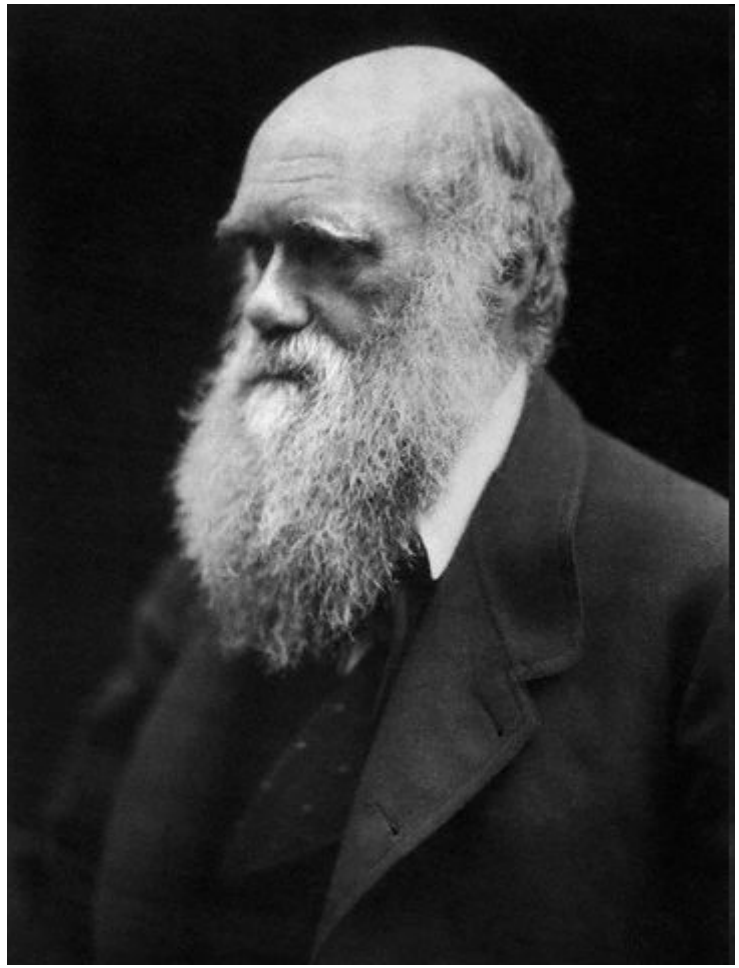
cryptography
Procedural Database Clients
OO Database Clients
Connectivity (> 10)
Filesystem
Operating system
Filesystem
Data formatting
caching
Image processing
XML processing (14)

Many more…..

Another view of PHP

- No specification
- Incomplete documentation
- Incomplete tests
- No Unicode
- Inconsistent implementation
  - Inconsistent naming and parameter order.
  - Bizare semantics in some corner cases.
- **Reference Implementation based language**

Is PHP a “proper” language?



Procedural and OO

```
<?php
   function  ($bean, $message=“default”) {
      $bean[“message”]=$message;
      return $bean;
   }
   ……..
   $b=foo ($mybean);
   $a=foo($mybean,”mymessage”);
 ?>
```

```
<?php
    class myclass {
       public $myProperty = ‘a default variable’;
       public function displayProperty() {
          echo $this->myProperty;
       }
    }
 ?>
```

Defaults to pass by value.

(easier for novice programmer)

Reference counting and copy-on-write allows efficiency.

Objects passed by reference in PHP5.

Stateless, shared nothing.

- Each “request” stands alone.
  - Think HTTP request.
  - –All program state rebuilt afresh on each request.
  - –State must be explicitly persisted.
    - Session (file, database, memcached....)
    - Database
    - File
    - Cached in the client(cookie, GET/POST parms.)

Dynamic Structure.

- Program structure is also rebuilt on each request.
- Definition of functions, classes, constants can vary from request to request.

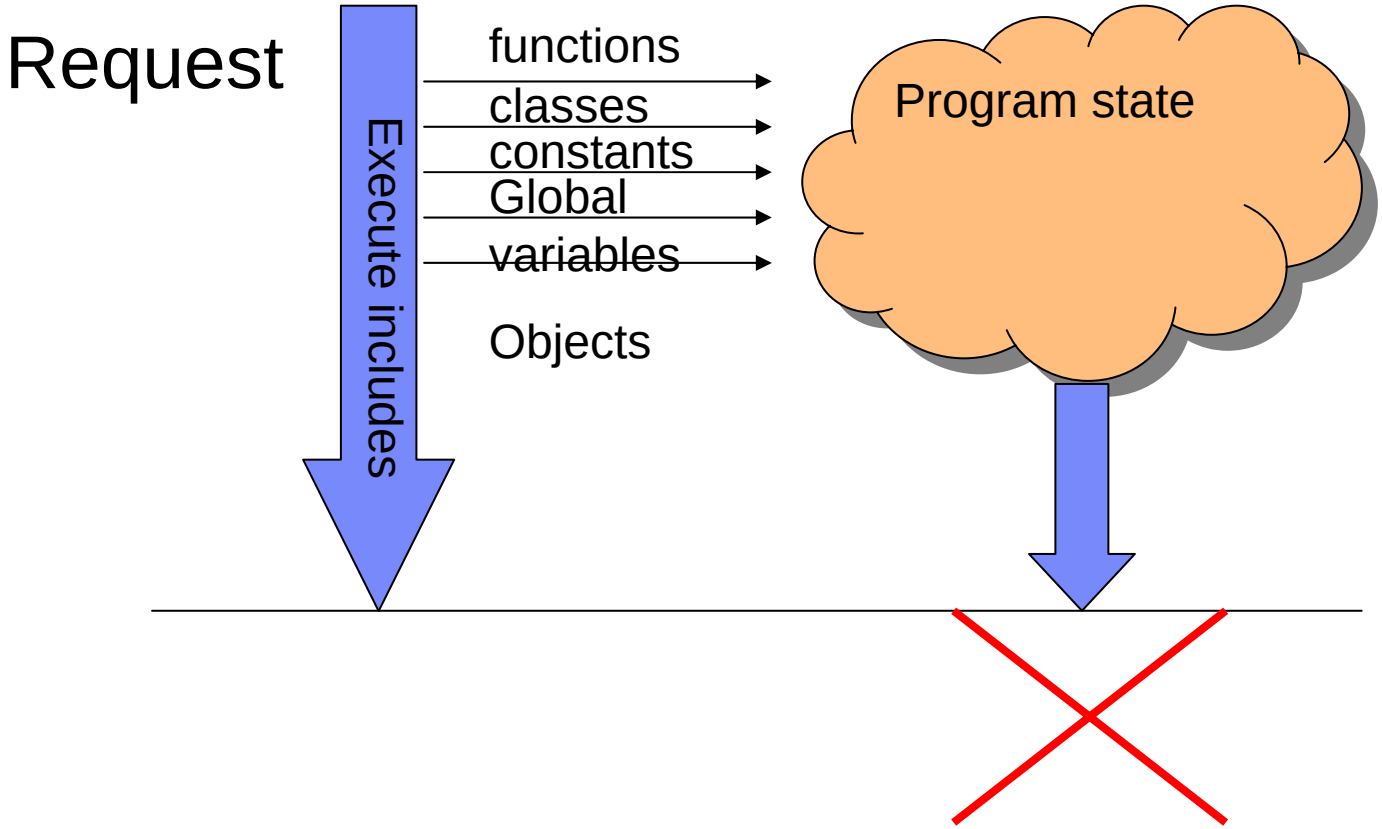
```
// filename: include1.php
<?php
   function foo ($a) {
      return ($a*2);
   }
 ?>
```

```
// filename: include2.php
<?php
   function foo ($a) {
      return ($a*4);
   }
 ?>
```

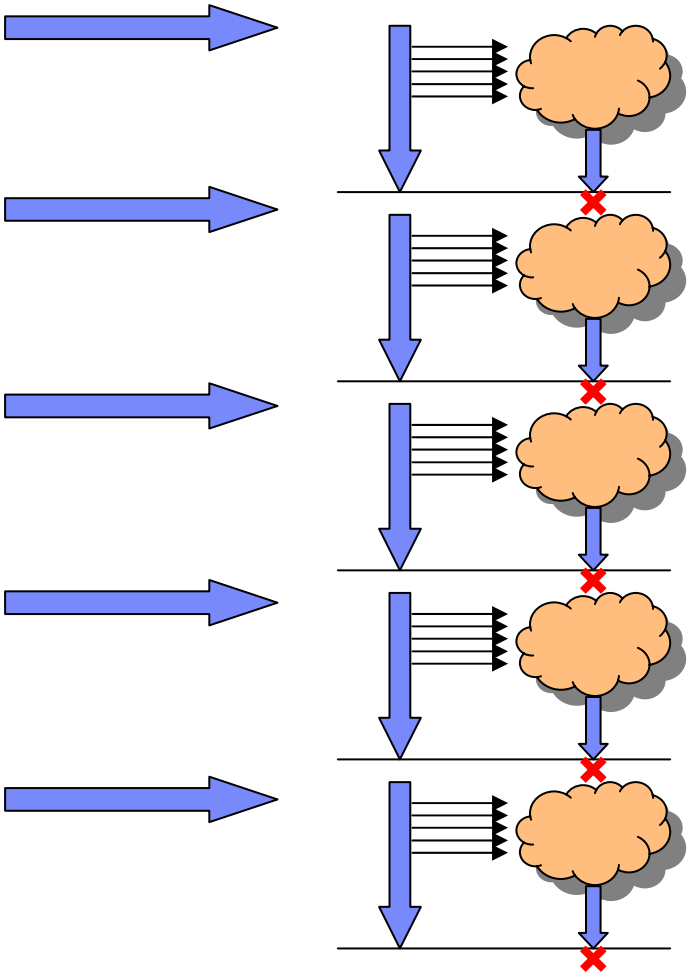
```
// filename: index.php
<?php
   if ($_GET[“myparm”] >2){
      include “include1.php”;
   } else {
      include “include2.php”;
   }
   echo foo ($_GET[“myval”]);
 ?>
```

Here the definition of the function foo changes based on a request parameter.

PHP Request Processing.



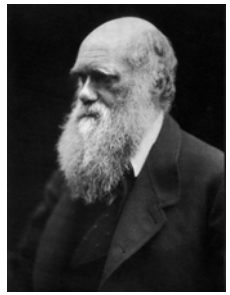
Shared nothing architecture



Threading model.

- Programming model has no threads.
  - No locks.
- PHP Programmer does not have to worry about
  - Thread safety.
  - Lock contention.
  - Deadlock.
  - Etc.
- Removes many sources of error and poor scalability.

Simple XML Handling in PHP



- PHP's SimpleXML.
  - 3rd Generation XML extension.
  - Represents XML document as PHP object tree.
- Set the value of an element:

```
$xml = new SimpleXMLElement($xmlstr)
$xml->movie[0]->characters->character[0]->name = 'Miss Coder';
```

- Set attribute "stars" on rating element using array syntax.

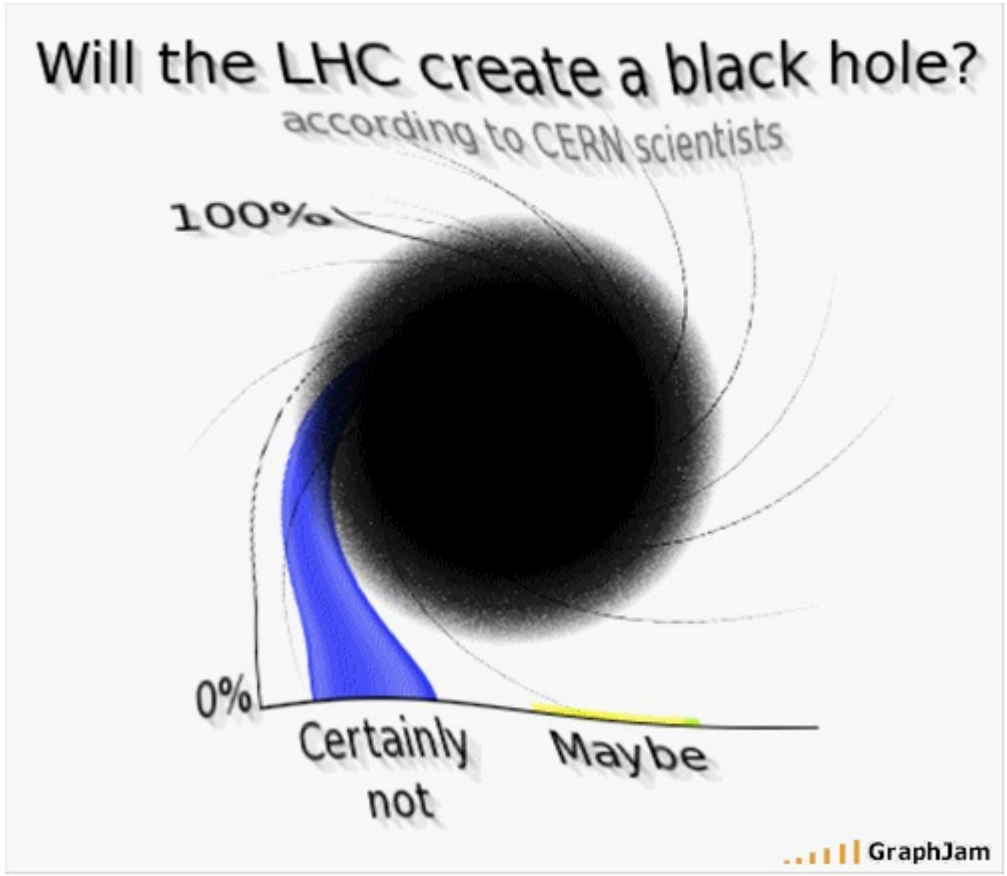
```
$xml = new SimpleXMLElement($xmlstr)
$xml->movie[0]->rating['stars']=3;
```

- SimpleXML also allows iteration.
- PHP also has a DOM extension and XML stream parsers.

A word about Unit test

Will the LHC create a black hole?

Posted: 29 Sep 2008 07:00 AM PDT



Graph by Joram H

Security

- Dynamic Language + unskilled programmers = security exposures.
- Early PHP implementations and applications were insecure.
  - Buffer overruns, CSRF, XSS, remote exploits.
- Language and implementation have evolved.
- Mantra:
  - Filter all input.
    - PHP has excellent “whitelist” based filtering.
  - Escape all output.
    - To the database and the client.
- Tainting has been proposed.

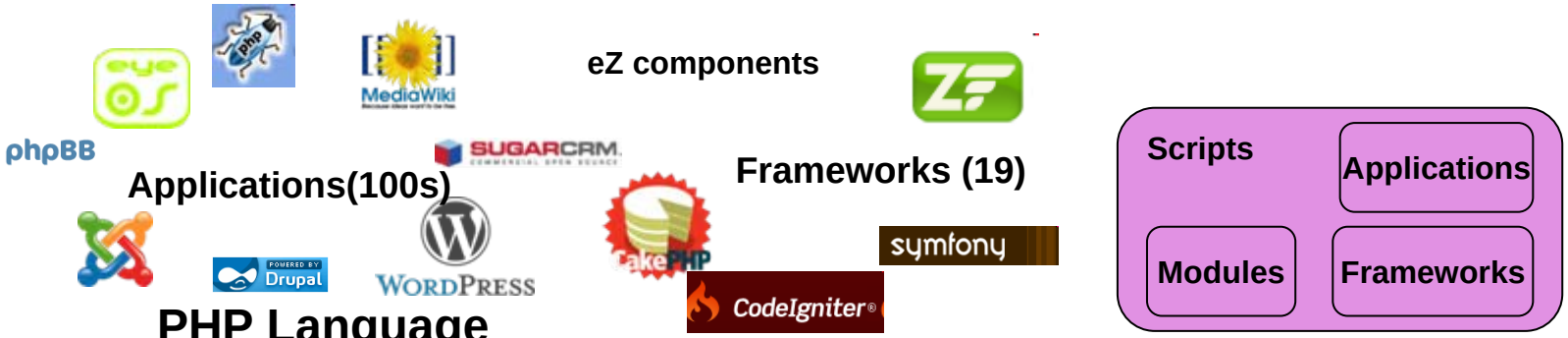
PHP Characteristics.

- Simple and Resilient
- Poor absolute performance
- Excellent scalability

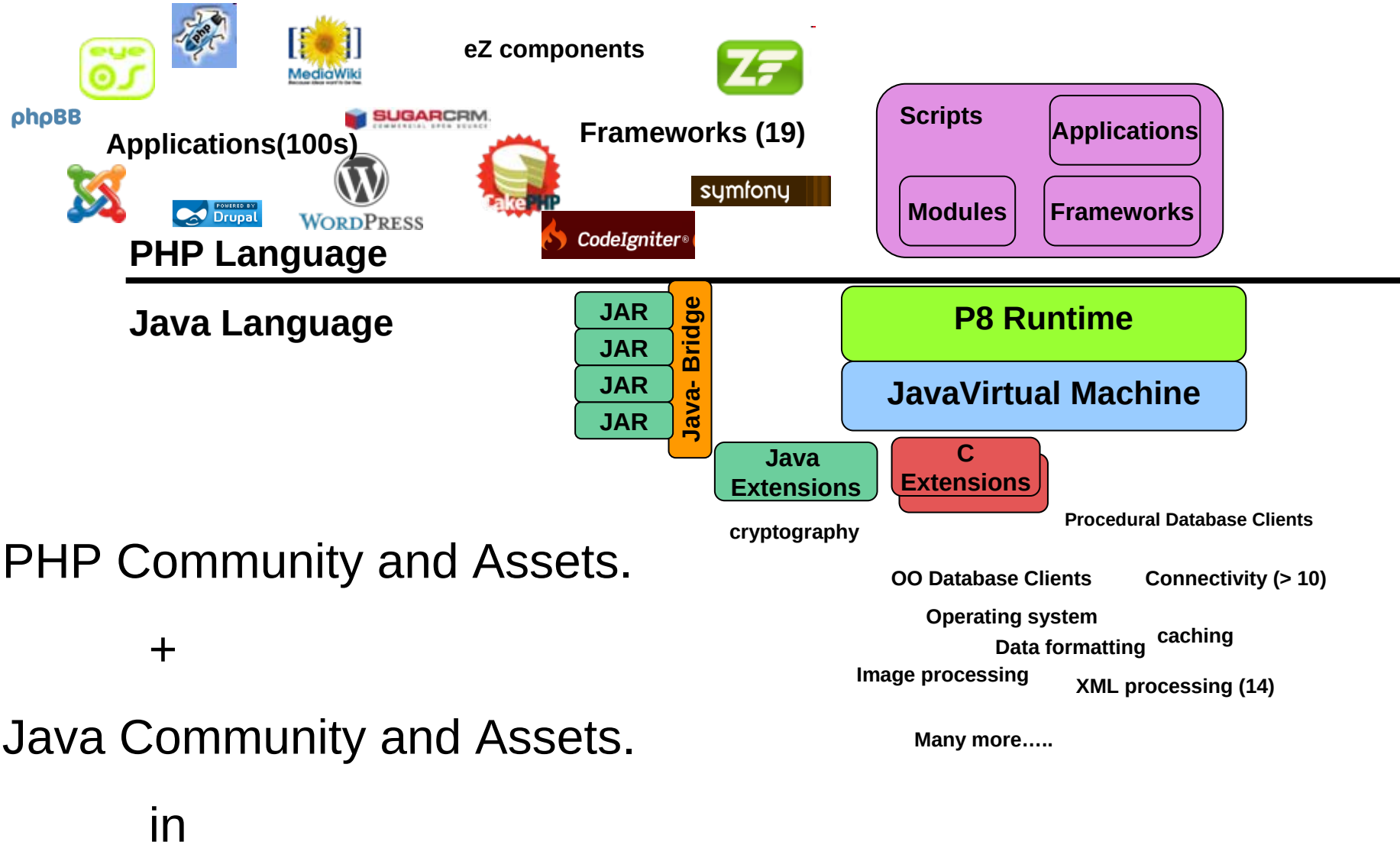
Agenda

- Why Scripting?
- Why PHP?
- What is PHP?
- Our PHP-on-Java implementation.
- PHP in CICS

PHP Reference Implementation



PHP-on Java Implementation Concept



PHP Community and Assets.

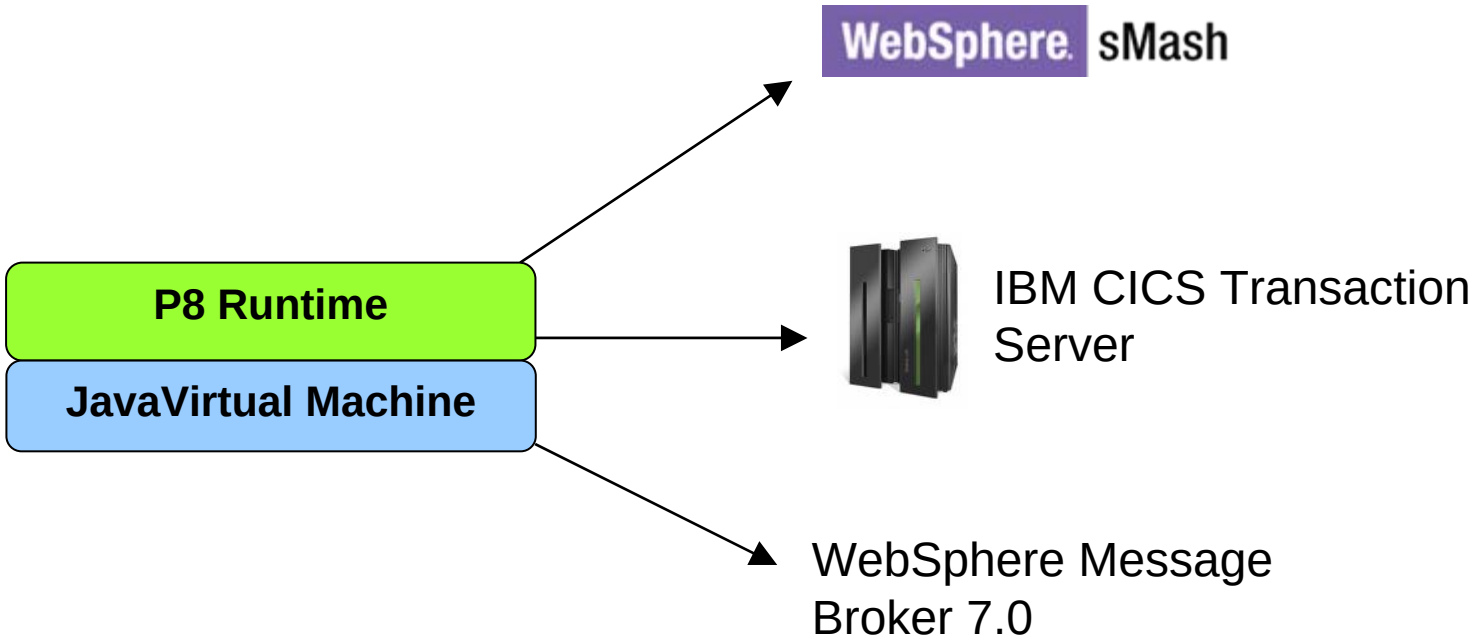
+

Java Community and Assets.

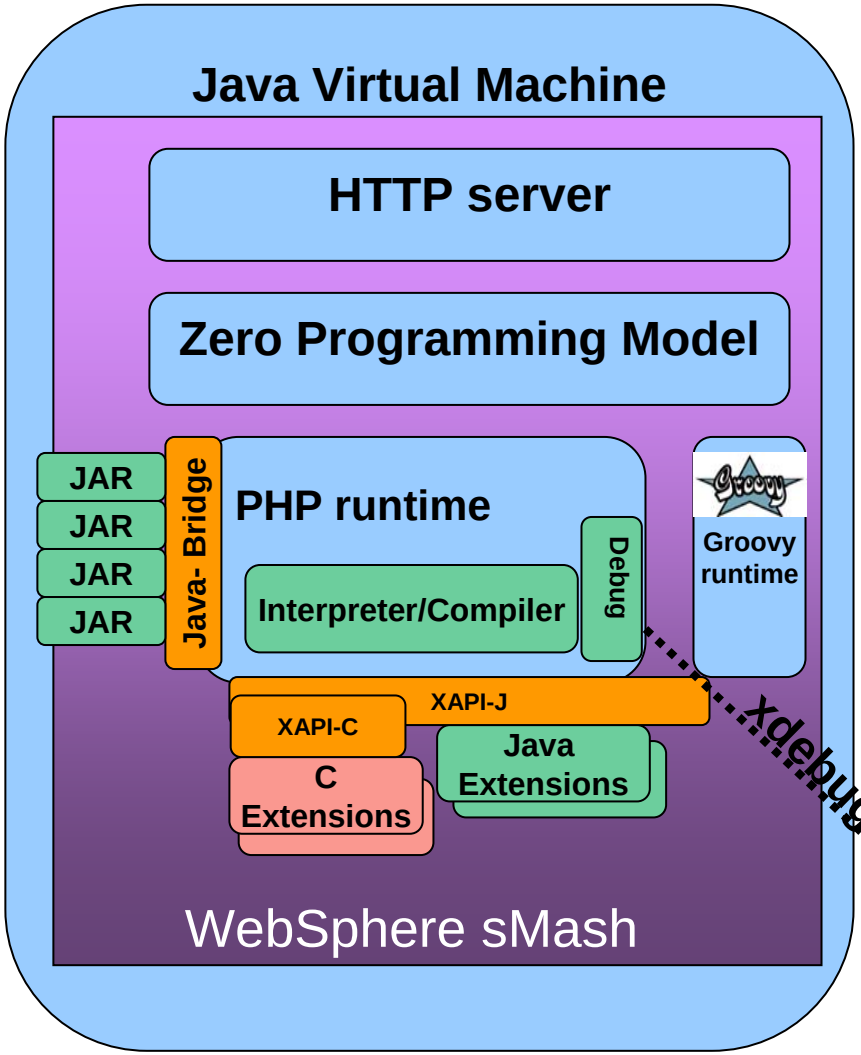
in

Java Platform environments

PHP in IBM Products.



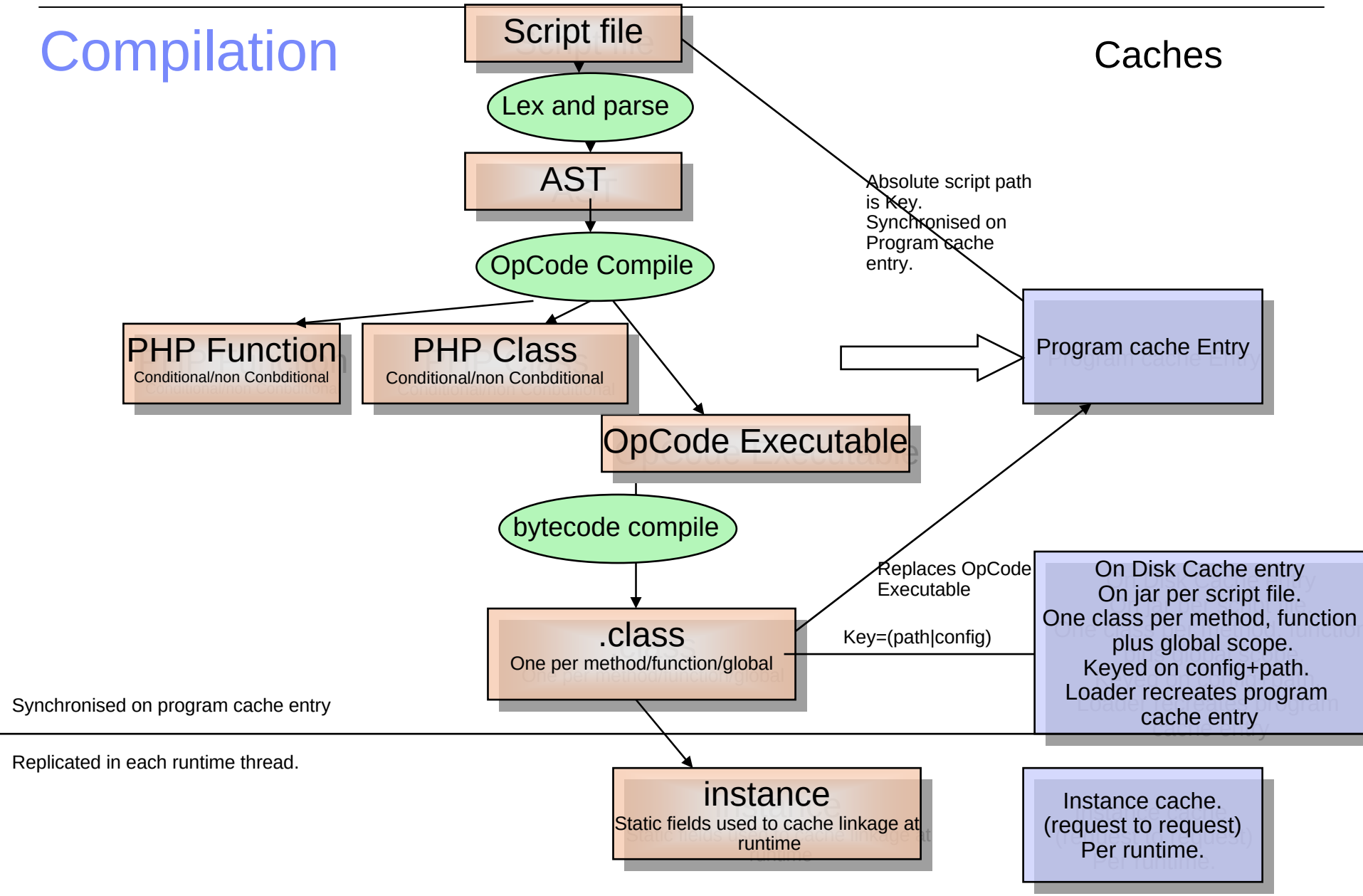
PHP in WebSphere sMash



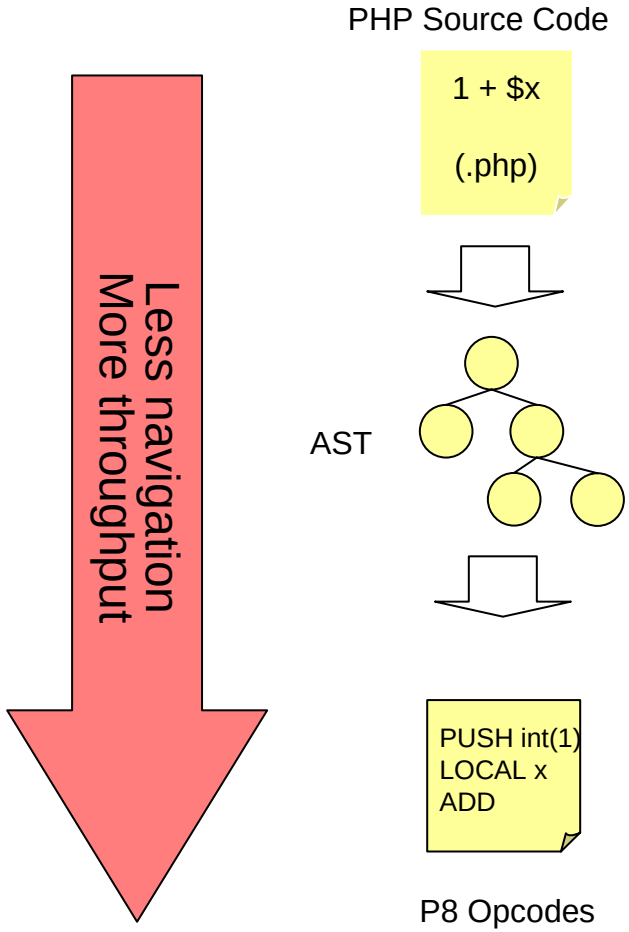
- Runs PHP 5 scripts
- Requires Java 5 SE or later.
- Extensibility via XAPI
  - XAPI-C for C extensions from php.net
  - XAPI-J for Java extensions, native libraries invoked over JNI and Project Zero interface
  - Extension language choice opaque to PHP script
- Java Bridge
- Debug using via xdebug protocol using Eclipse with PDT

STOP!

Compilation

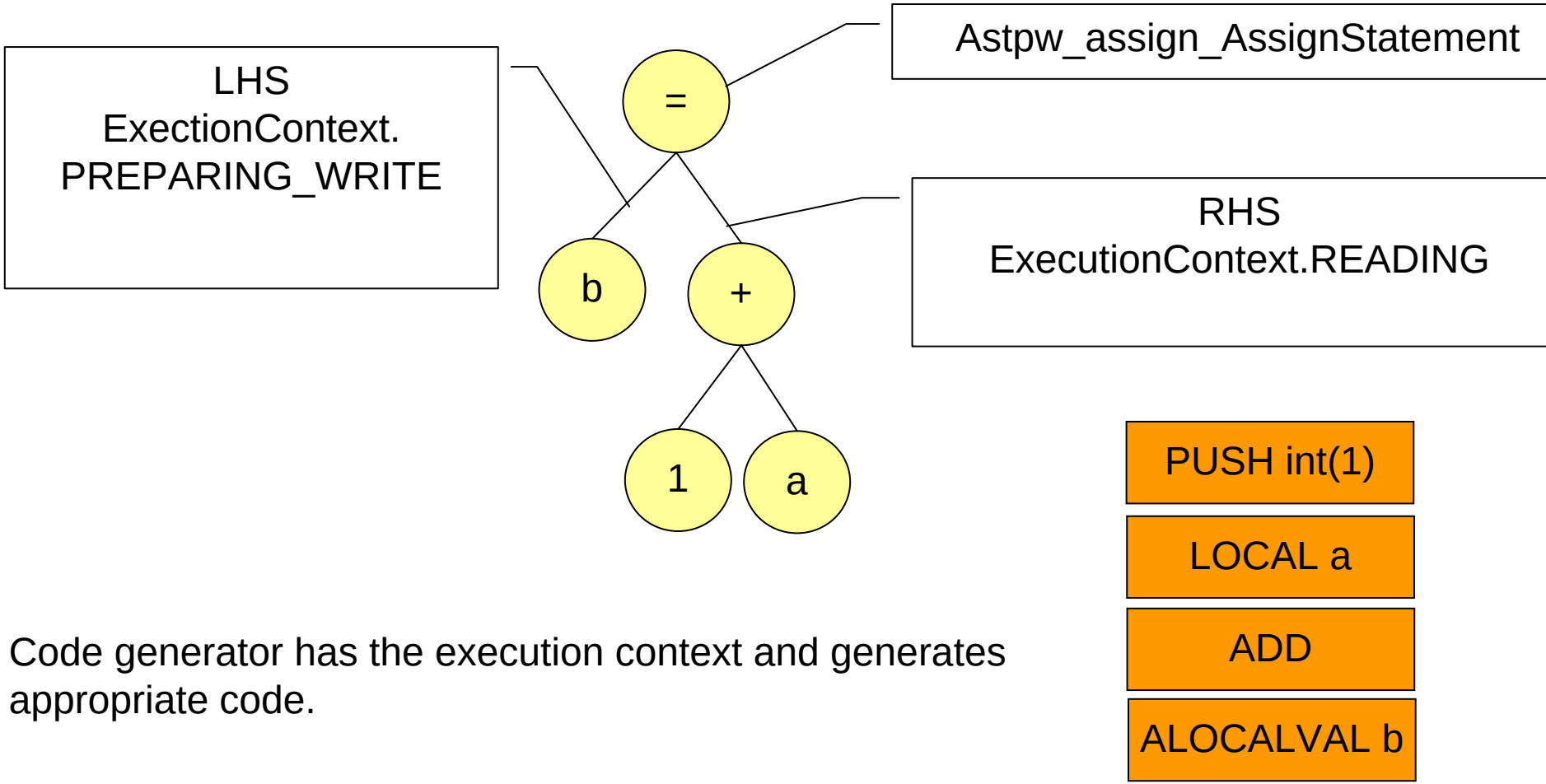


Code Generation



- AST descended recursively to generate flat list of opcodes.
- Opcodes are unique to P8 and are stack based which is key to later translation to Java bytecode.
- Some simple optimisation are done. Execution context is evaluated and appropriate code generated.

Execution Context



Code generator has the execution context and generates appropriate code.

PHP evaluation order

```
$a[a1()][a2()];
```

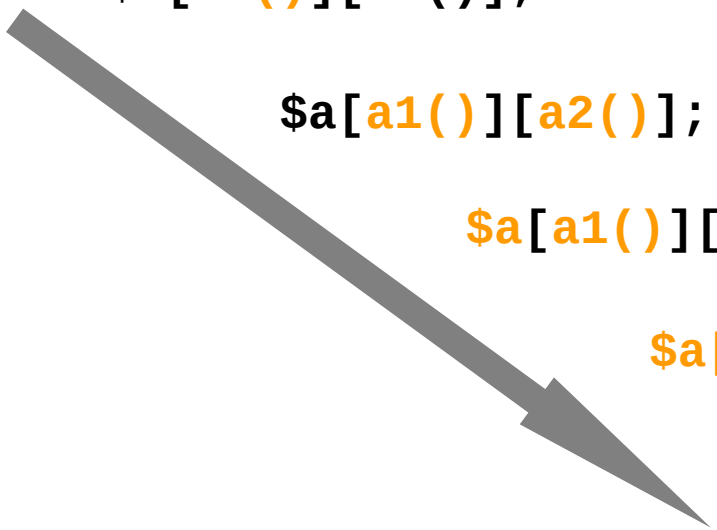
```
$a[a1()][a2()];
```
Call a1

```
$a[a1()][a2()];
```
Call a2()

```
$a[a1()][a2()];
```
Get $a

```
$a[a1()][a2()];
```
Index into a by result of a1()

```
$a[a1()][a2()];
```
Index into a[a1()] by result of a2()



Further Execution Order Examples

- **`$a[a1()][a2()] = $b[$b1()][$b2()] = $c[c1()][c2()];`**
  - –Order is a1(), a2(), b1(), b2(), c1(), c2(), c[][], assign b[][], assign a[][]

- **`$a[a1()][a2()] = $b[$b1()][$b2()] + $c[c1()][c2()];`**
  - –order is a1(), a2(), b1(), b2(), evaluate $b[][], c1(), c2(), evaluate c[][], assign a[][]
  - –Could this be an unintentional inconsistency? … bug? Tested on 5.2.1

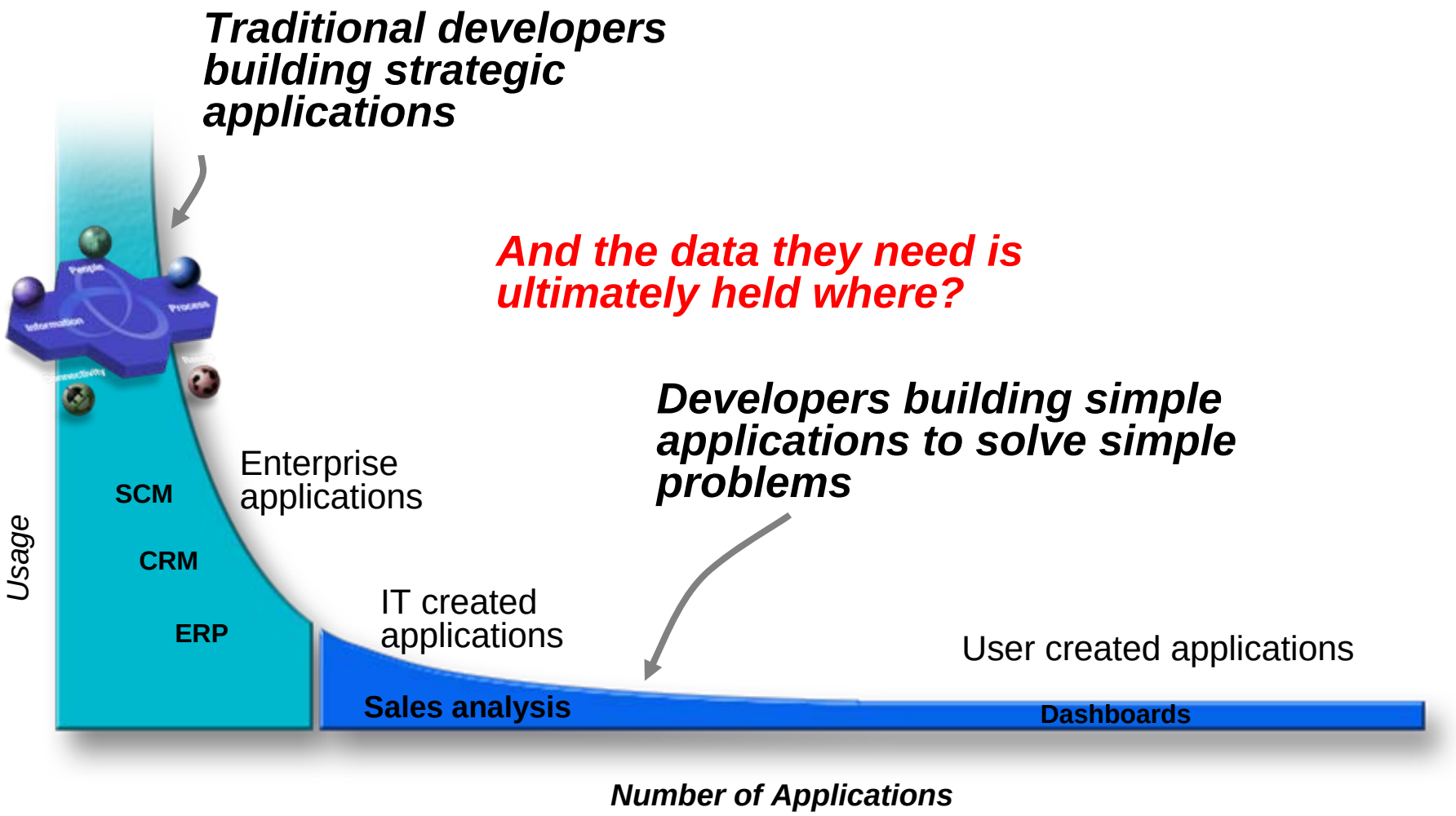
Agenda

- Why Scripting?
- Why PHP?
- What is PHP?
- Our PHP-on-Java implementation.
- PHP in CICS

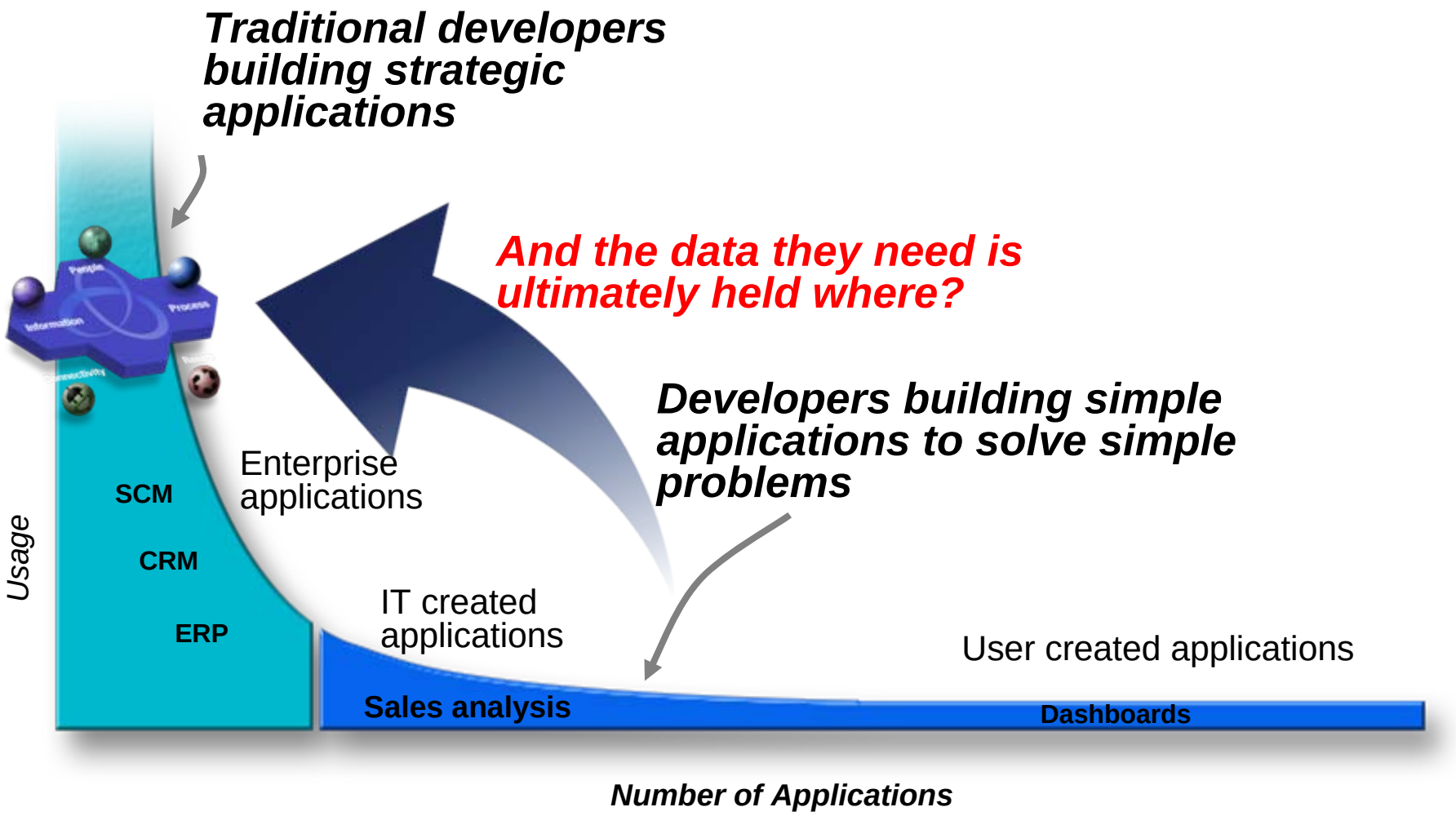
OLTP meets scripting... why?

- CICS systems execute millions (billions?) of business critical transactions per day
  - –The “ities” - Secure, reliable, available, …
  - –Assembler, COBOL primarily
- Has evolved many invocation styles/technologies
  - –3270 “green screens”
  - –SNA, APPC
  - –MQ Series
  - –TCPIP
  - –HTTP
  - –SOAP

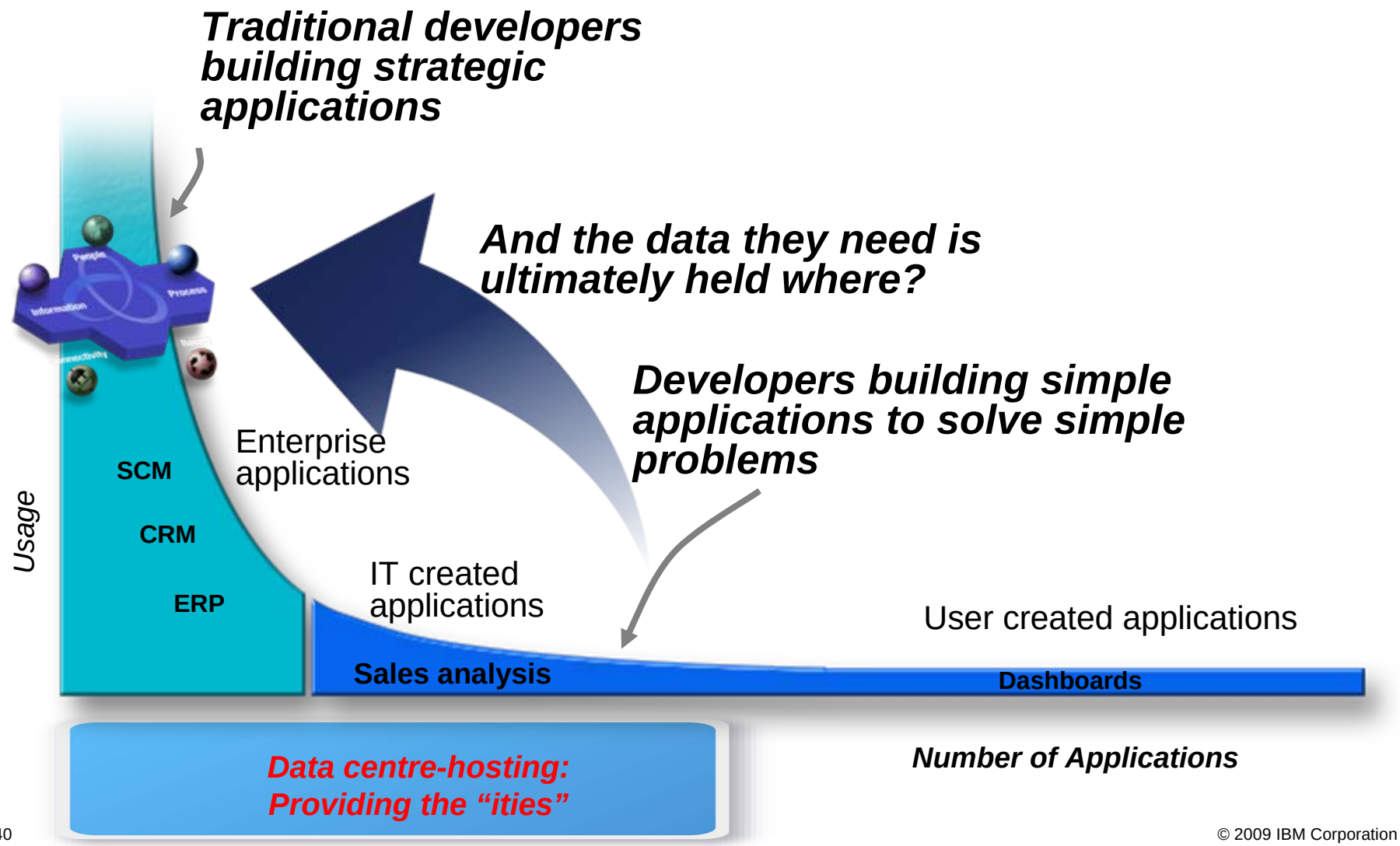
The Application Landscape



The Application Landscape



The Application Landscape



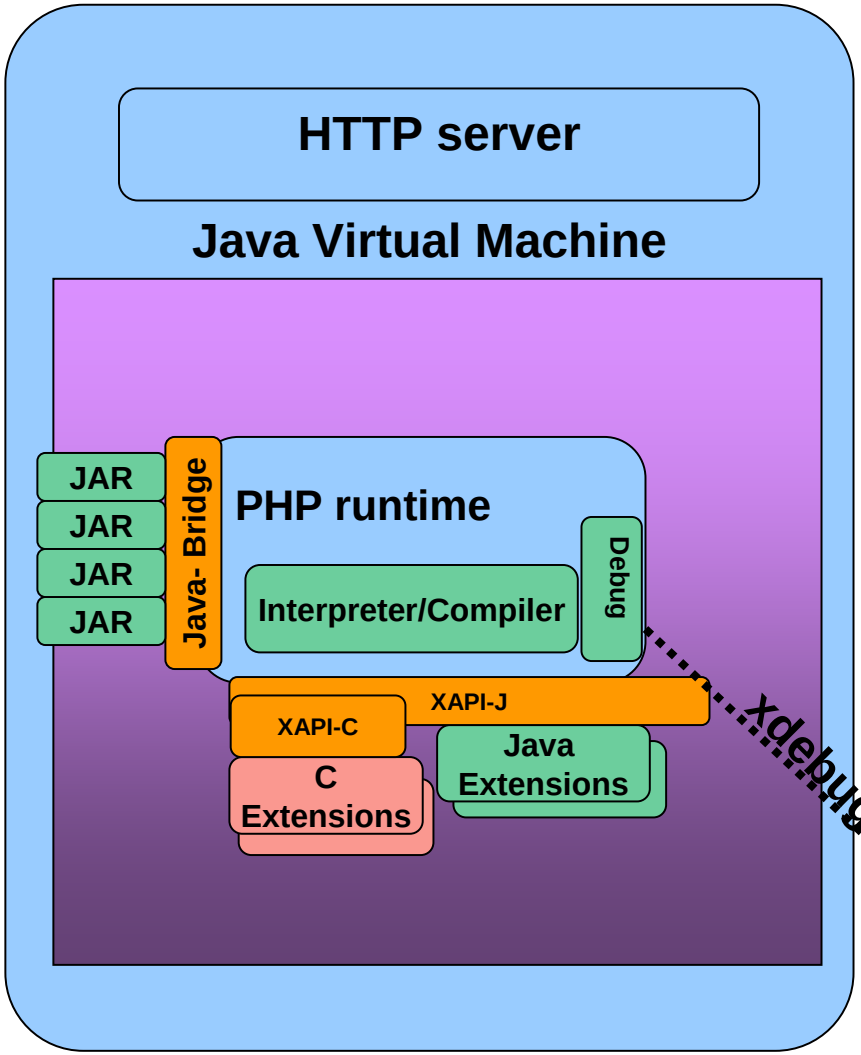
Web programming with PHP and CICS

- Basic CICS/COBOL programming pattern is very similar to PHP
  - Request/Response
  - No threading
  - No implicit persistence
  - CICS Pseudo-conversations == the first shopping cart!
- Many CICS apps fit a RESTful model eg Atom
  - Find collections of data
  - View a collection
  - Pick an item
  - Update it

What is REST ?

- REST is the acronym for „**Re**presentational **S**tate **T**ransfer“
- It is the architectural model on which the World Wide Web is based
- Principles of REST
  - Resource centric approach
  - All relevant resources are addressable via URIs
  - Uniform access via HTTP – GET, POST, PUT, DELETE
  - Content type negotiation allows retrieving alternative representations from same URI
- REST style services
  - are easy to access from code running in web browsers, any other client or servers
  - can serve multiple representations of the same resource
- More info: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

PHP in CICS Transaction Server

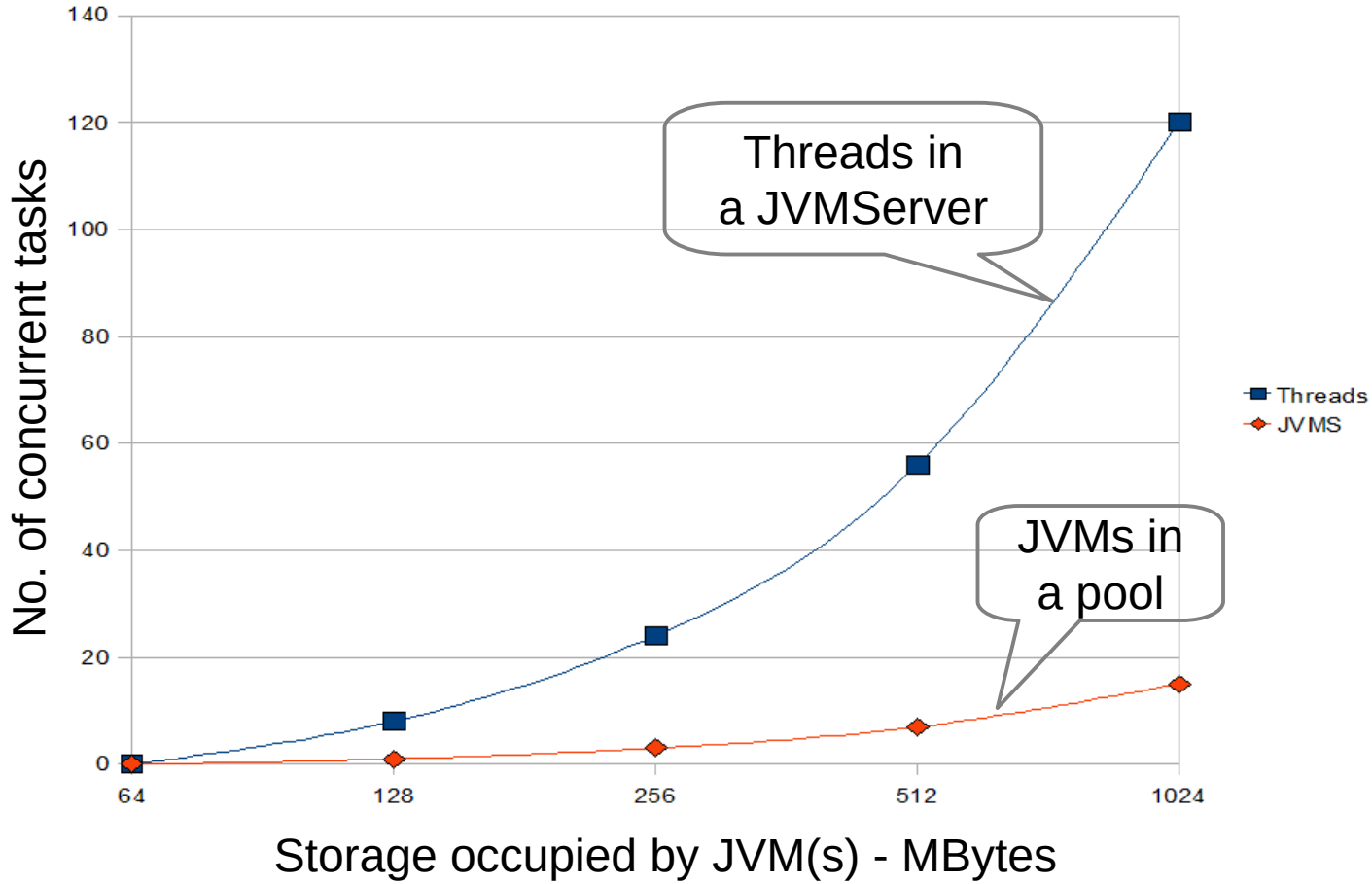


- Runs PHP 5 scripts
- Requires Java 5 SE or later. CICS v3.2
- Extensibility via XAPI
  - XAPI-C for C extensions from php.net Subset compiled for System z
  - XAPI-J for Java extensions, native libraries invoked over JNI and Project Zero interface
  - Extension language choice opaque to PHP script
- Java Bridge. Access to CICS and DB2
- Debug using via xdebug protocol using Eclipse with PDT

JVMServers

- JVM implementation in CICS continues to evolve
  - JDK 1.1.8
  - HPJ and Hotpooling
  - IBM Persistent Reusable JVM (Shiraz)
  - Continuous mode
  - Java 5
  - Java 6
- Now, JVMServers

Comparing capacity (projected)



Graph assumes 18 Meg for Base JVM size, + 40 Meg of Engine, statics, classes etc, + 8M of app storage usage per thread.

(Null thread – no application state of its own, and not causing any addition classes to be loaded = 40Kb per thread)

JVMServers - why?

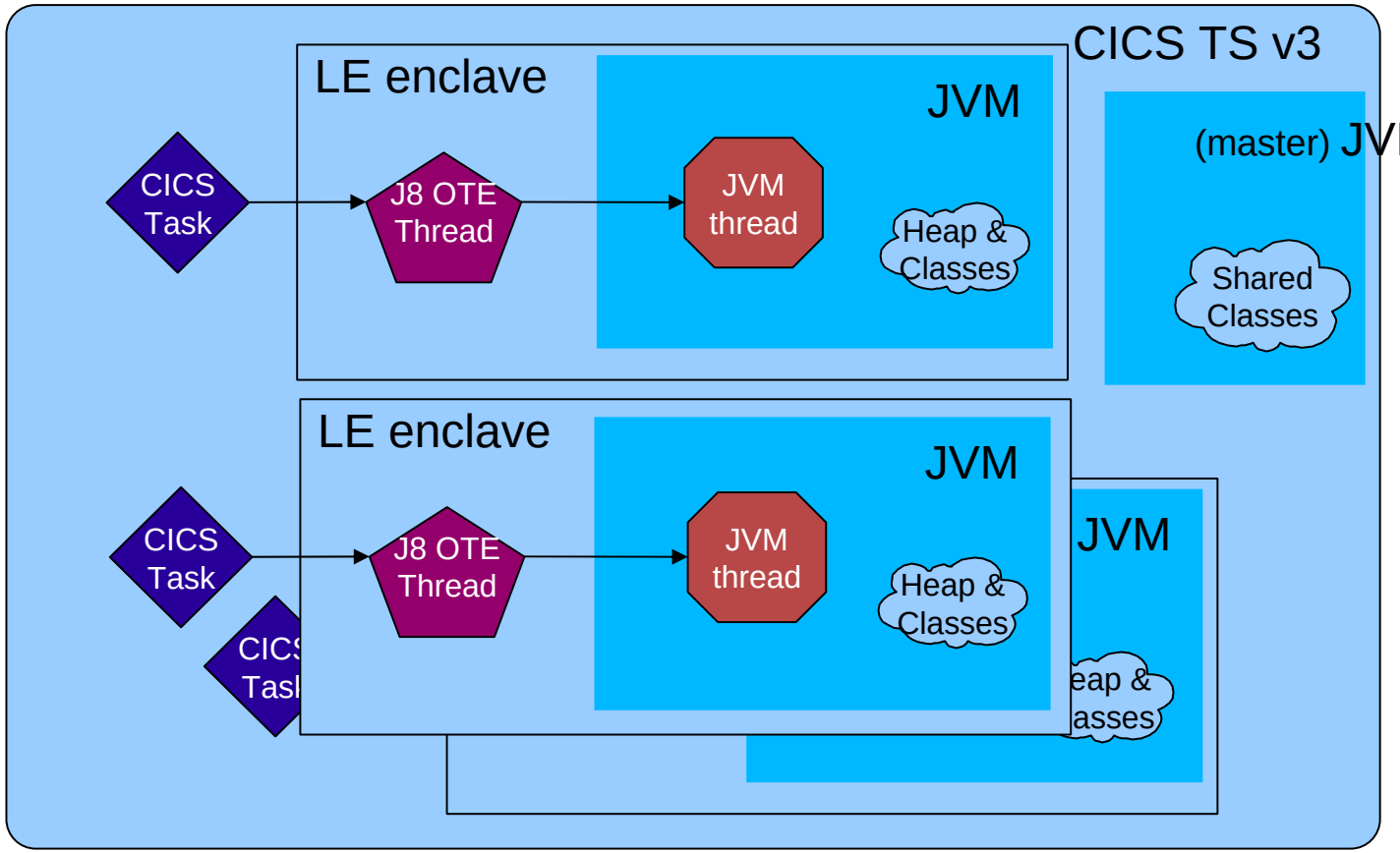
- JVMs up to now
  - Single task, serial reuse
  - Large memory footprint
  - Excellent isolation characteristics
- JVMServers
  - Multiple tasks (threads) in a JVM concurrently
  - Larger capacity
  - Risk of collateral damage
  - Not for customer application use in v4.1

JVMPool Architecture - CICS TS v3 (and v2)

Single CICS task dispatched into a JVM in the pool at a time. So concurrent task count limited to the number of JVMs that can fit in the region.

Each JVM 'costs' ~20Mb plus the application heap value.

Result is about 20 task/JVMs concurrently in each region.

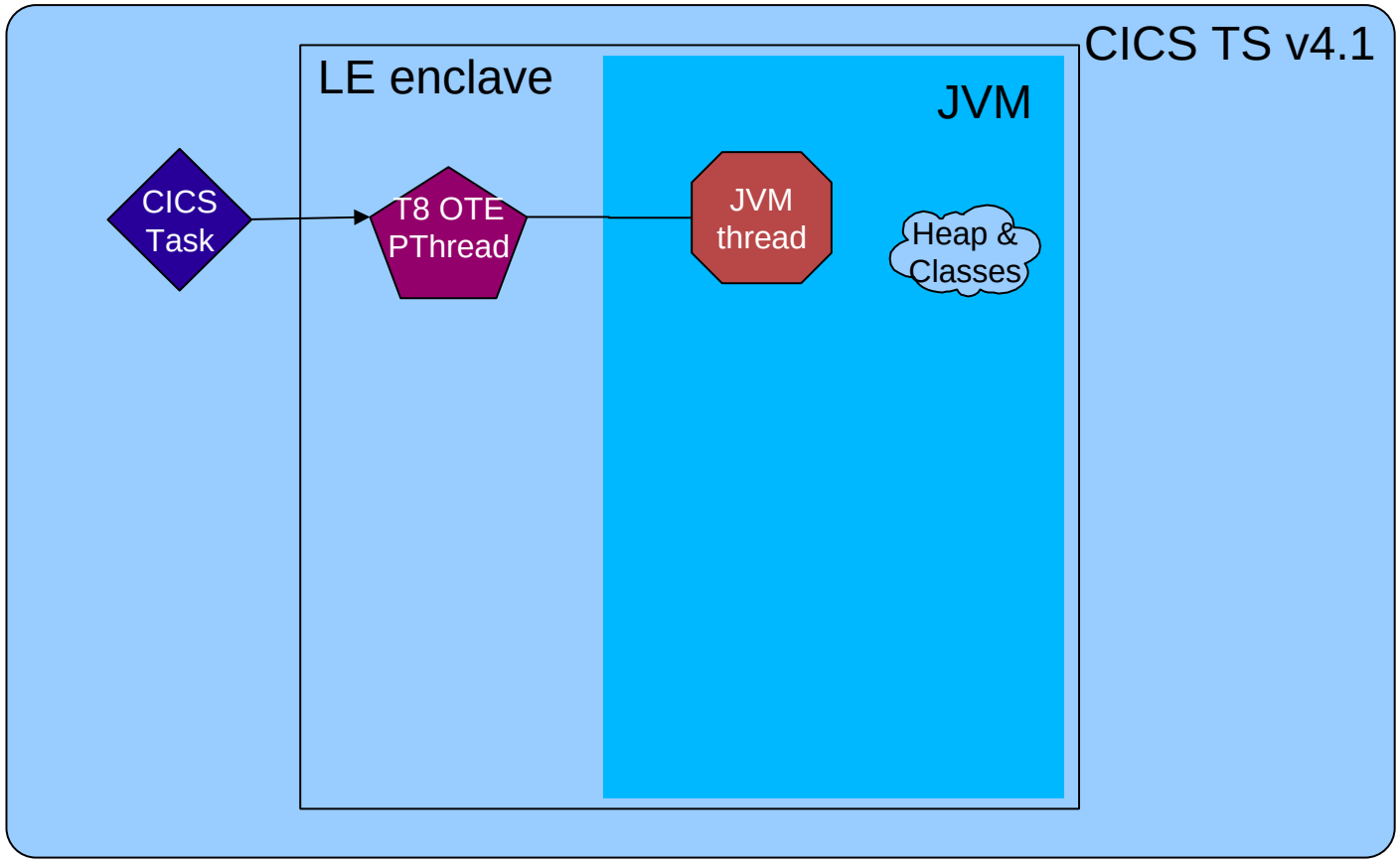


JVMServer Architecture

New CICS TCB "mode".

Called "T8" - dubbed as both a CICS TCB and an LE "pthread".

JNI call to attach a pthread to an existing JVM.



JVMServer Architecture

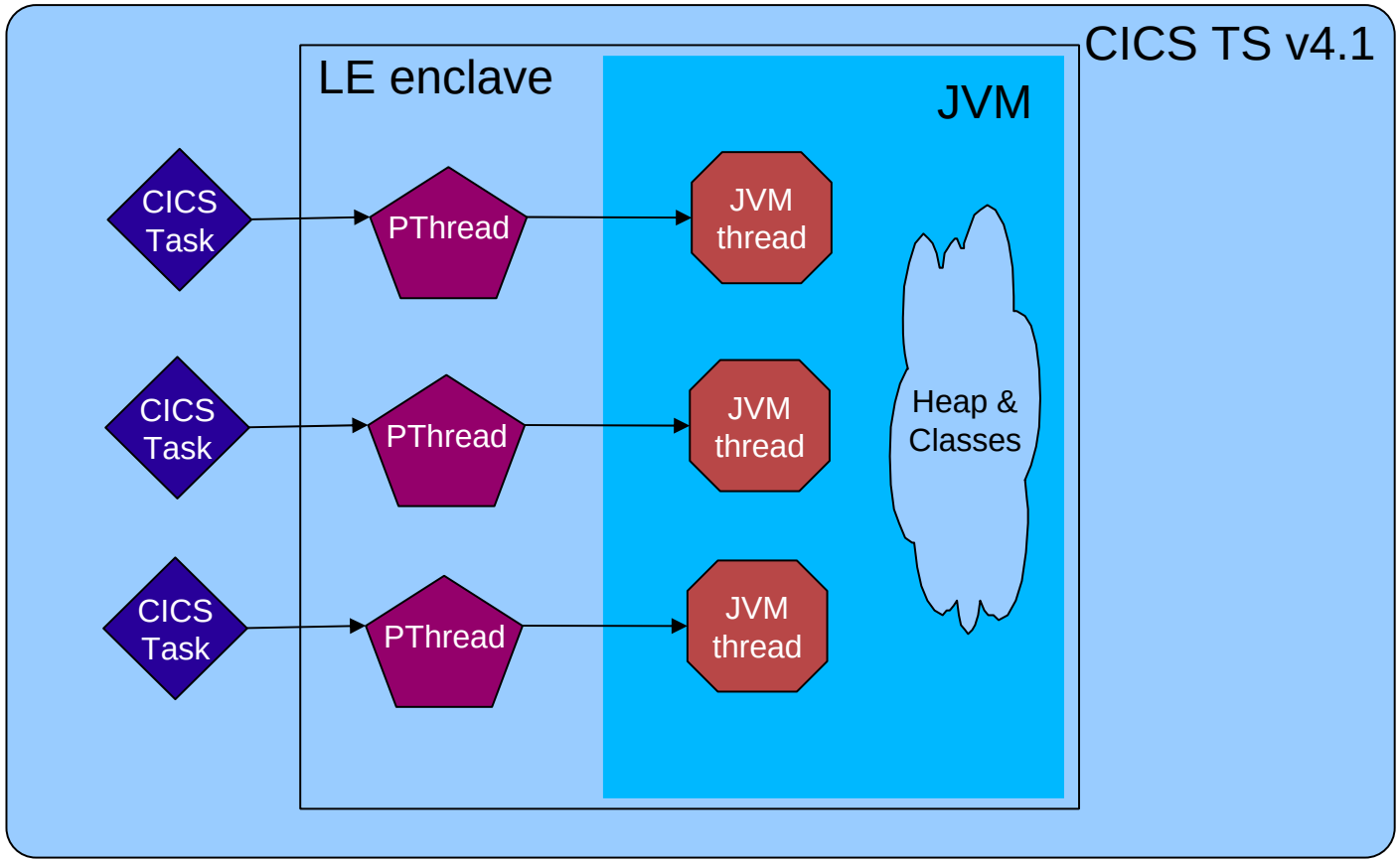
Can attach multiple pthread/T8/CICS tasks to the JVM at the same time.

Therefore serve more requests using a single JVM.

JVMServer thread "cost" is...

Very, very like a WAS servant region.

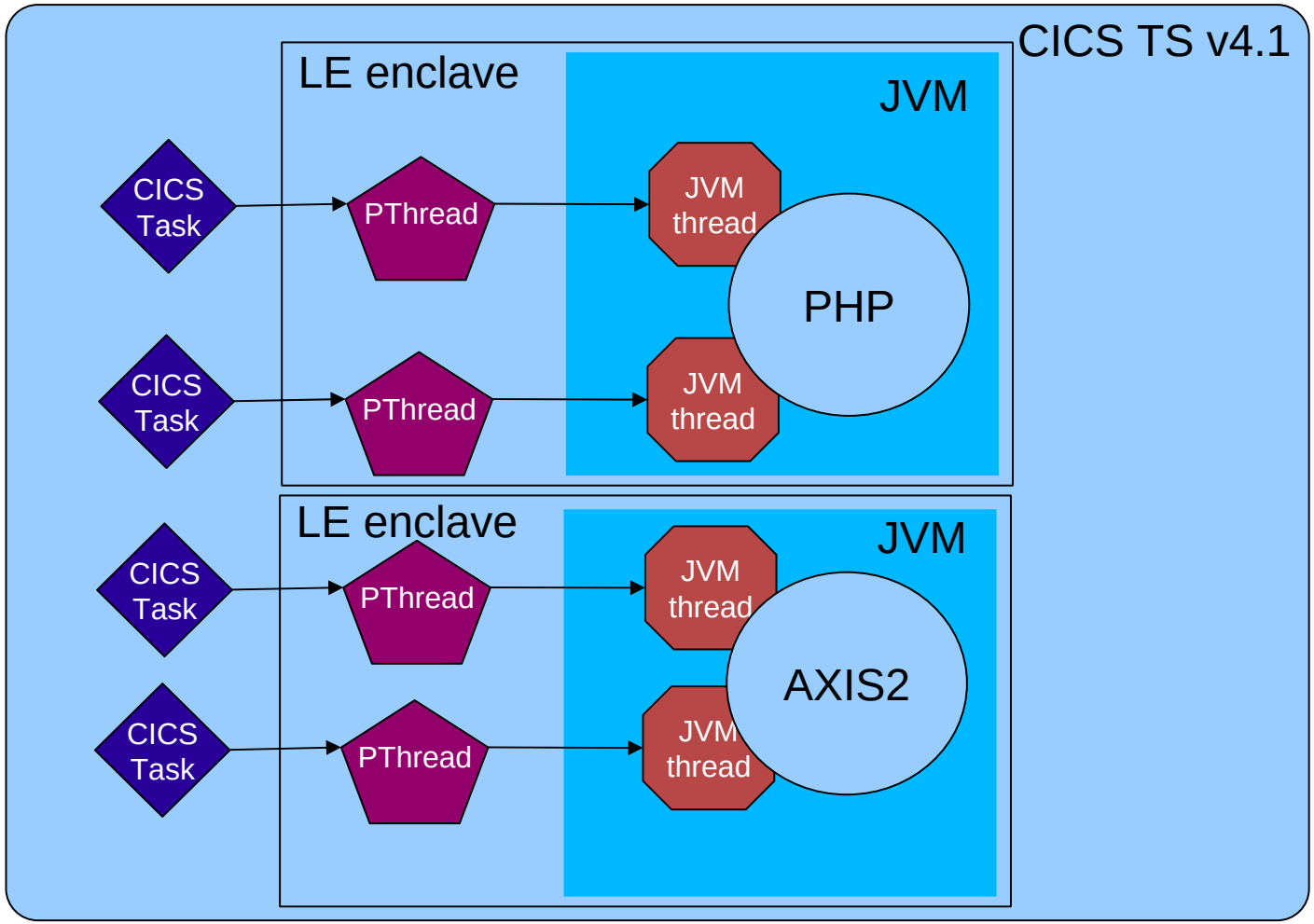
Result is .... tasks per region.



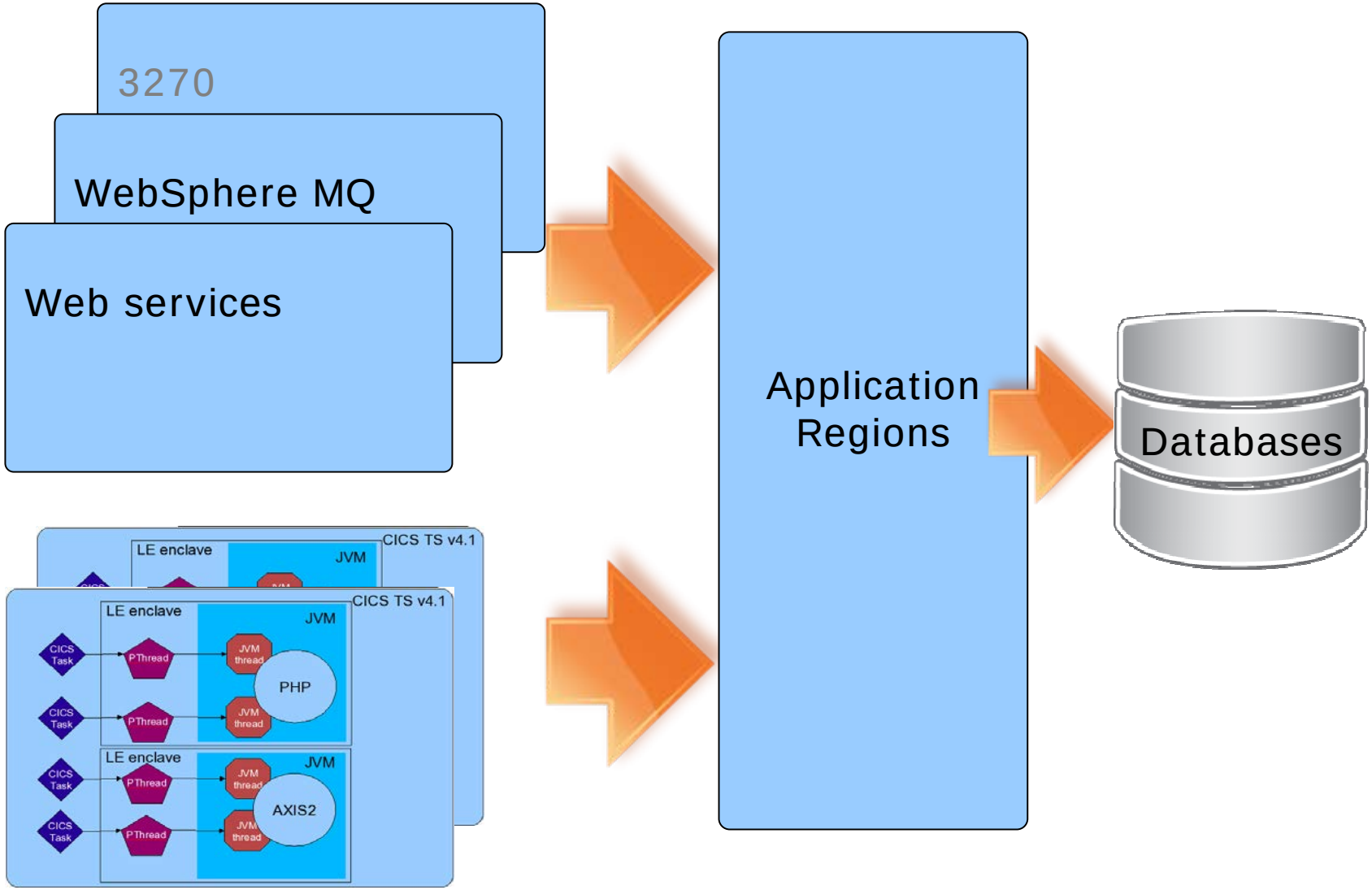
JVMServer Architecture

Architected to allow multiple JVMServers in a single CICS.

Different types of work, or just a degree of isolation.



Deploying PHP for agility



In Conclusion

Simplicity wins.

Reliable simplicity wins reliably.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries:
ibm.com/legal/copytrade.shtmlAIX, CICS, CICSPlex, DataPower, DB2, DB2 Universal Database, i5/OS, IBM, the IBM logo, IMS/ESA, Power Systems, Lotus, OMEGAMON, OS/390, Parallel Sysplex, pureXML, Rational, Redbooks, Sametime, SMART SOA, System z , Tivoli, WebSphere, and z/OS.

A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office
Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.