Mono Meeting.

Miguel de Icaza
miguel@novell.com

October 24, 2006

Novell®

Mono, Novell and the Community.

Mono would not exist without the community:

- Individual contributors.
- Companies using Mono.
- Organizations using Mono.
- Companies using parts of Mono.
- Google Summer of Code.

Introductions.

Goals of the Meeting.

A chance to meet.

- Most of the Novell/Mono team is here.
- Many contributors are here.
- Various breaks to talk.

Talk to others!

- Introduce yourself, ask questions.

Talk to us!

- Frank Rego, Mono's Product Manager is here.
- Tell us what you need in Mono.
- Tell us about how you use Mono.

Project Status

Goals

Originally:

- Improve our development platform on Linux.

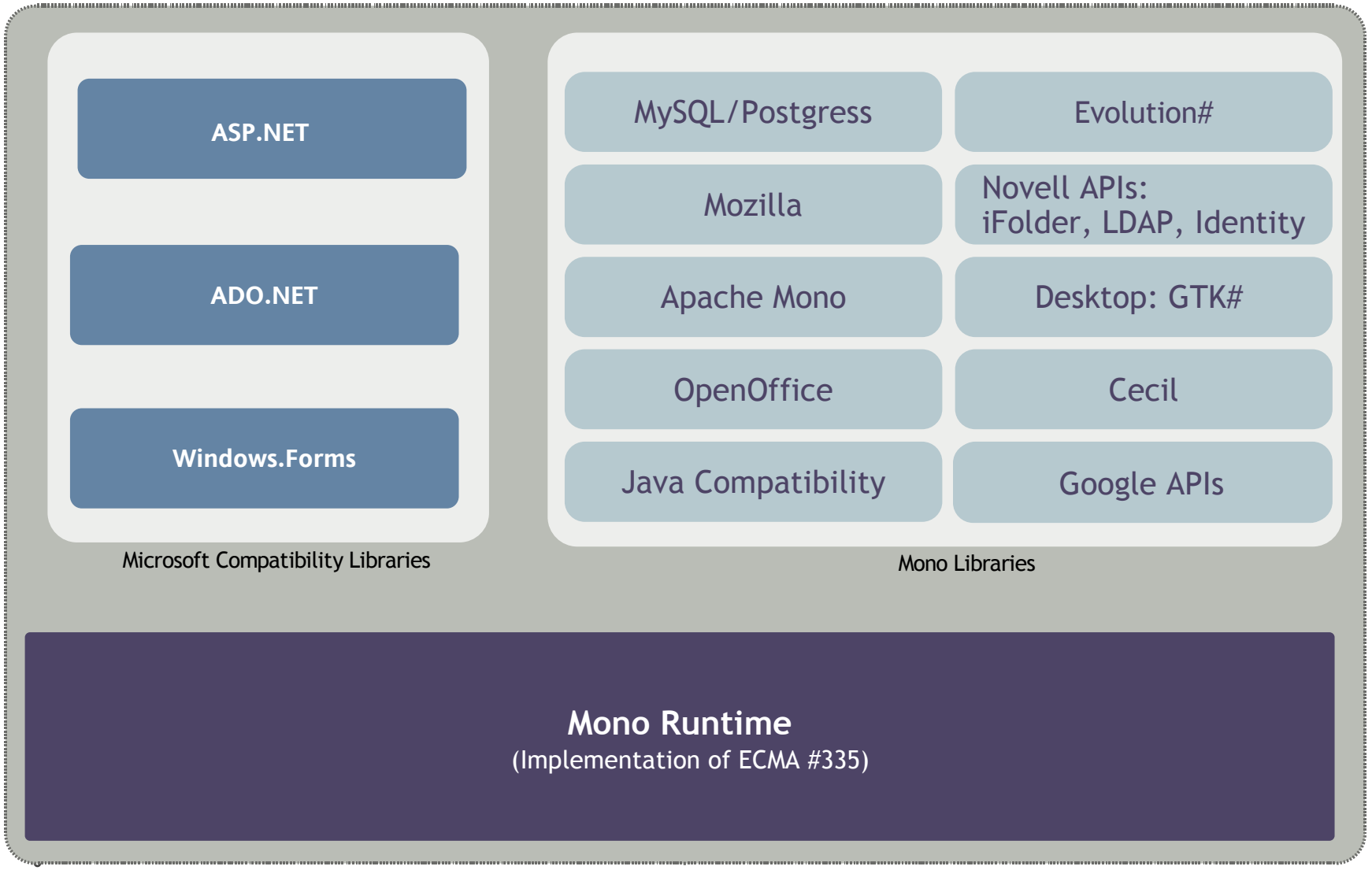
As the community grew:

- Expand to support Microsoft APIs.

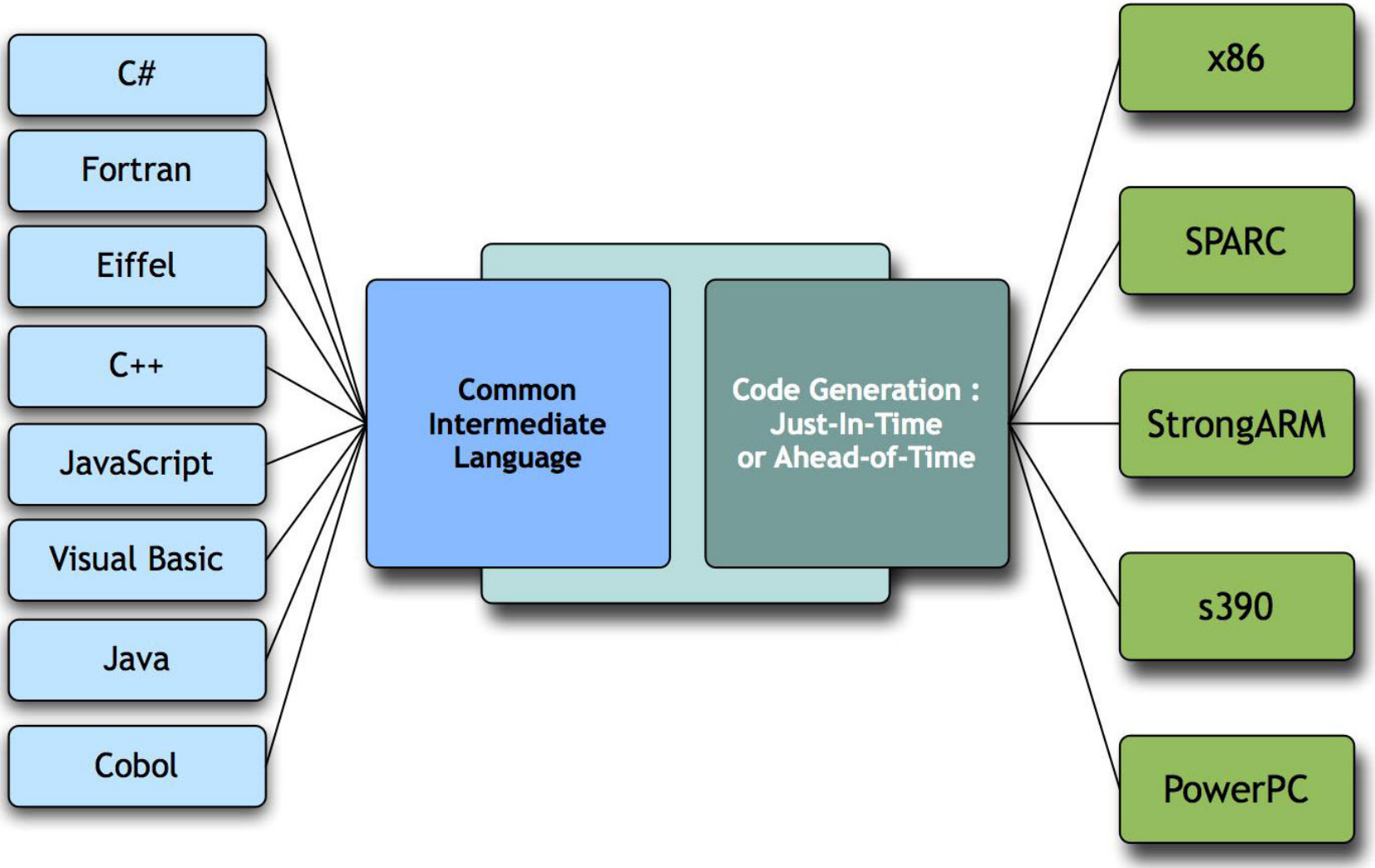
As Mono got more complete:

- Provide a complete cross platform runtime.
- Allow Windows developers to port to Linux.

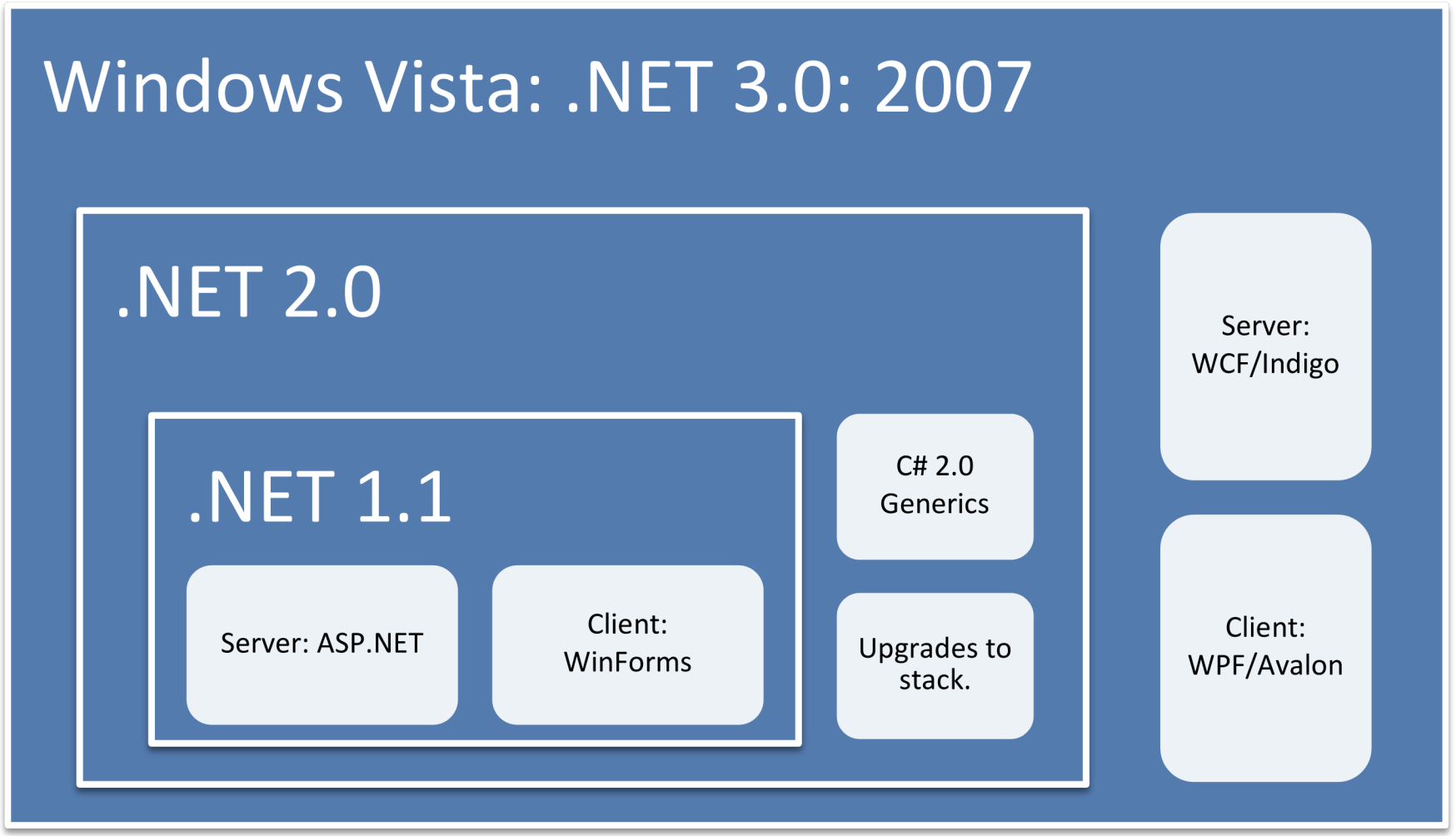
Mono Stacks and Goals.



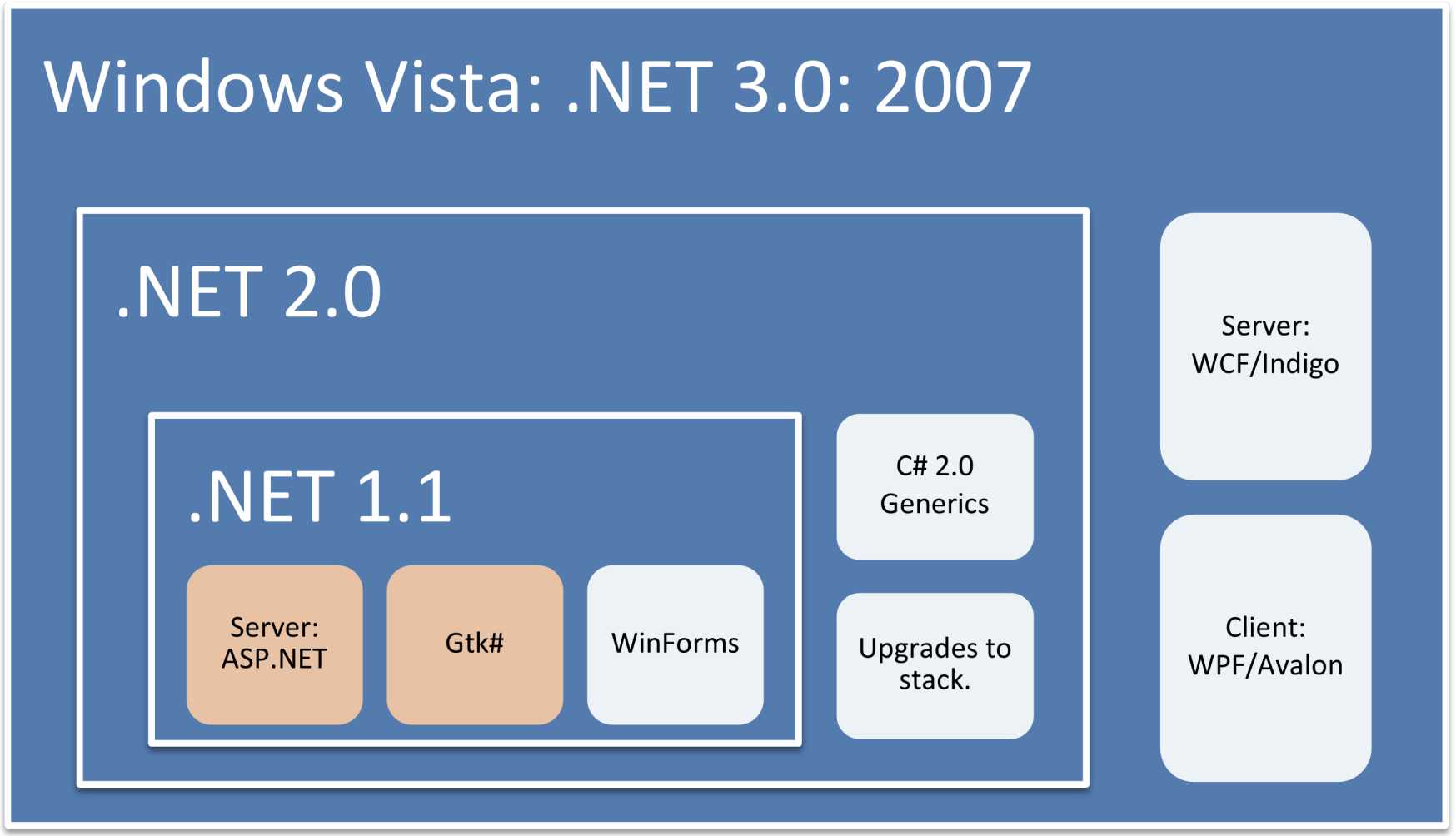
Platforms, CIL, Code Generation.



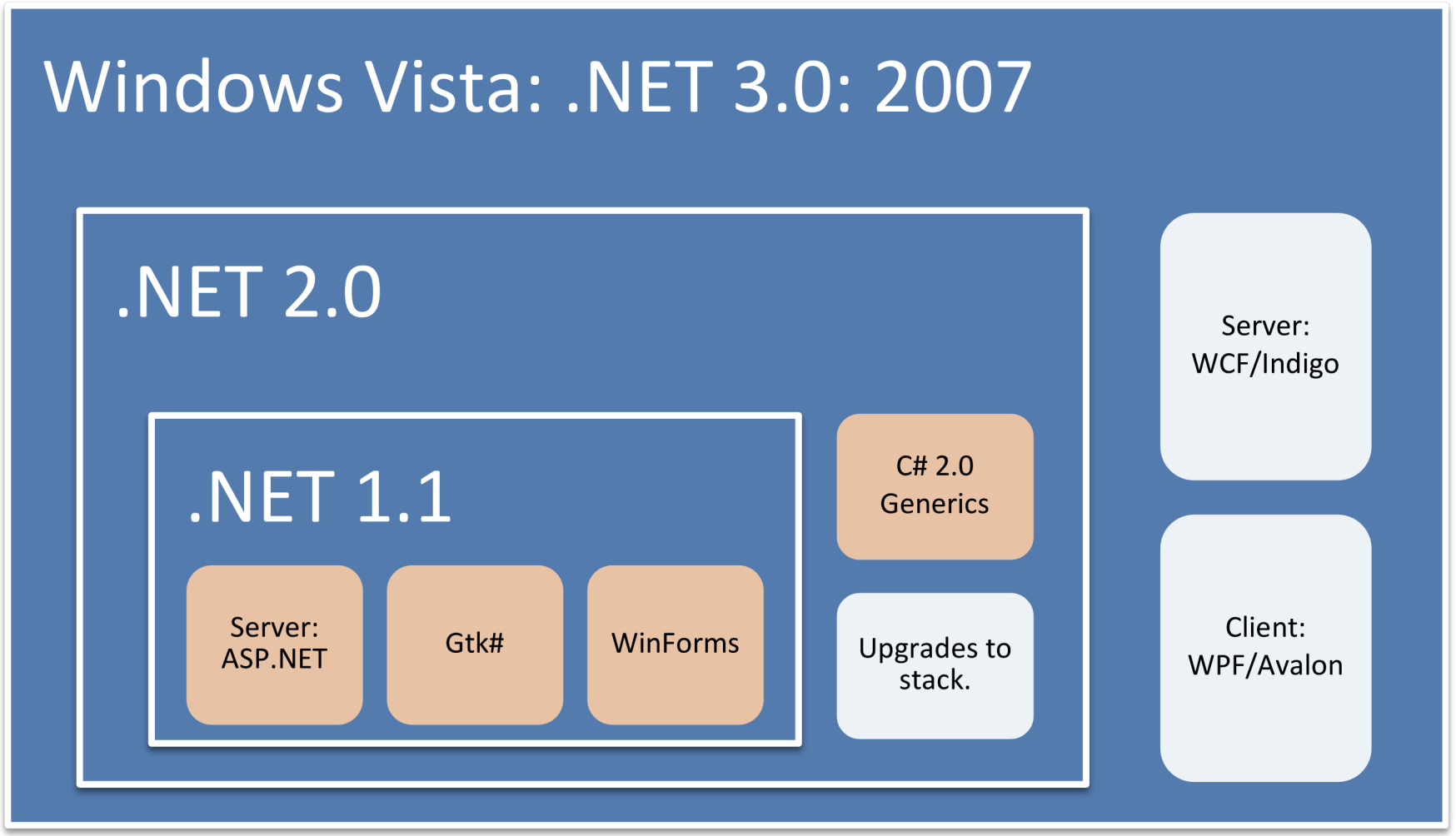
API space


Windows Vista: .NET 3.0: 2007
.NET 2.0
.NET 1.1
Server: ASP.NET
Client: WinForms
C# 2.0 Generics
Upgrades to stack.
Server: WCF/Indigo
Client: WPF/Avalon

Mono 1.0: July 2004
"T-Bone"



Mono 1.2: November 2006
“Rump steak”



Mono 1.2 bits.

Reliability and scalability:

- ZenWorks and iFolder pushed Mono on the server.
- xsp 1.0: 8 request/second.
- xsp 1.2: 250 request/second.

GUI

- Windows Forms 1.1 debuts.
- Gtk# 2.x series: updated binding, updated to Gtk+ 2.8

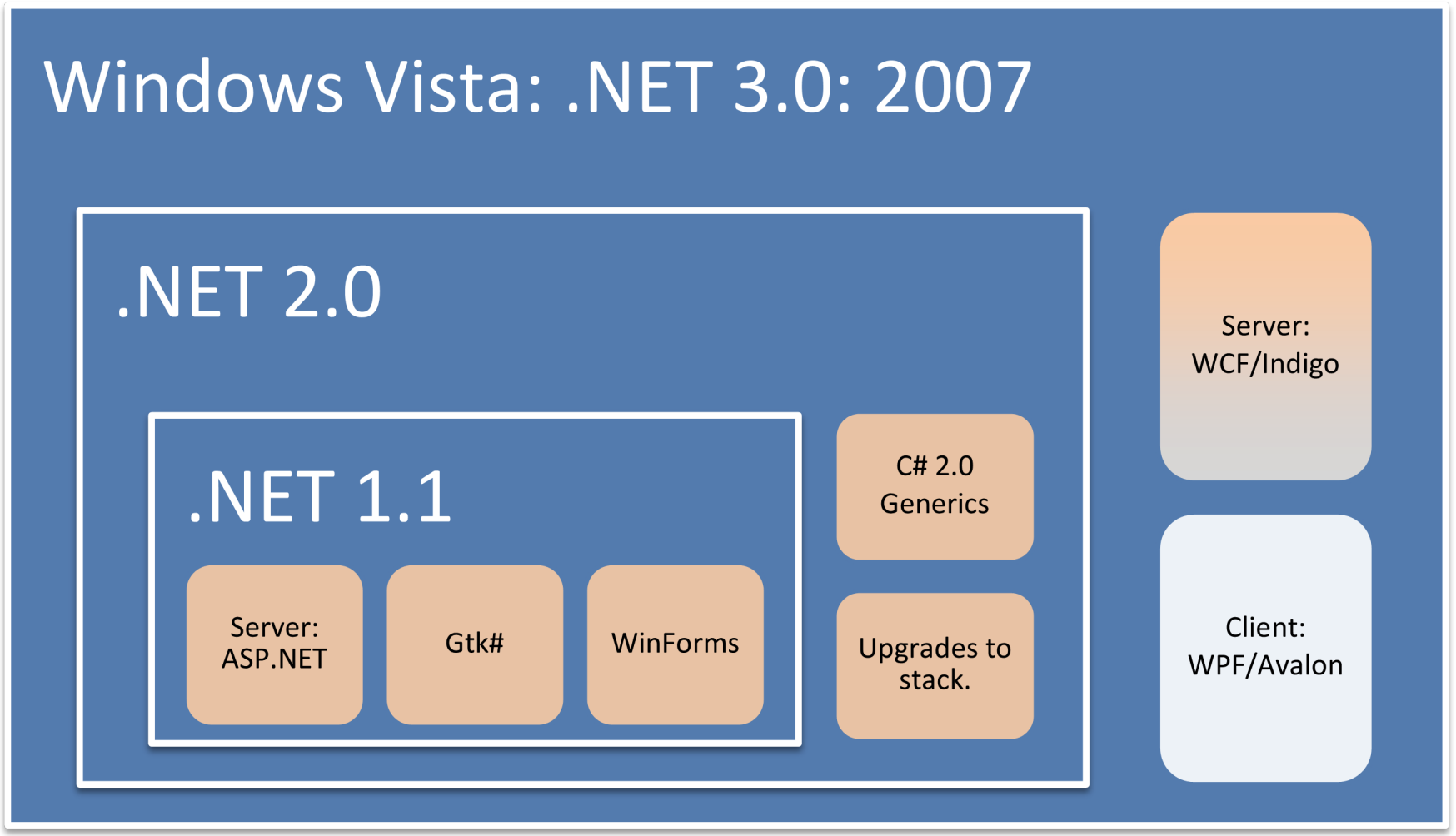
C# 2.0, .NET 2.0

- Complete.
- With VM support.
- Some 2.0 API support.
- IronPython works.

Debugger:

- x86 and x86-64 debugger.
- CLI-only, limited in scenarios (no xsp).
- Needs usability testing.

Mono 2.0: Q3 2007
“Sirloin”



Sirloin Directions.

Mono 2.0: Core.

- .NET 2.0 API support.
- CAS available.
- New optimizations.
- Compacting GC.
- MonoDevelop.
- MonoDevelop + Debugger.

Improve Support:

- Windows integration, build.
- Visual Studio integration.
- MacOS X and X-Code.

Gtk#

- Databinding support.
- Others (Mike's talk).

Languages:

- Ship Rolf's VB compiler
- GCC-based compilers.

Announcement
Olive.

Olive Project

Under development, not ready, not done.

- Contributions, as always, welcomed.

Today we release:

- Basic Indigo implementation.
- Basic Infocard implementation.

Previously done:

- System.Query, System.Xml.XLinq
- System.Workflow
- System.Windows.Serialization (Xaml support and `xamlc`).

Implementing an API.

Based on MS documentation.

- Documentation is sometimes incomplete, not clear.
- Might be missing details.
- The programmer might not understand things.

Test-based implementation:

- Write NUnit test case to explore the API
- Make the test run on Windows.
- Serves as blueprint for Mono implementation.

Tests in Mono.

As of September, 2006:

Class Library Tests:

- 91,000 feature tests for
- 2,227 classes

Compiler tests:

- 1,100 positive tests
- 1,500 negative tests

Scripting

Scripting: Higher Level Programming.

John Ousterhout: “ Scripting: Higher Level Programming for the 21st Century”, 1998:

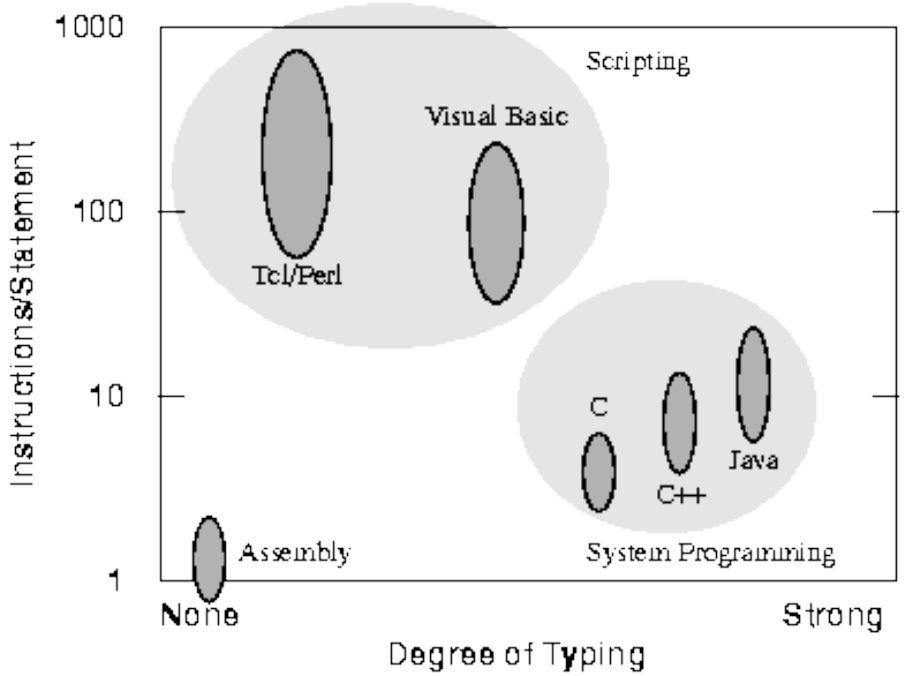


Figure 1. A comparison of various programming languages based on their level (higher level languages execute more machine instructions for each language statement) and their degree of typing. System programming languages like C tend to be strongly typed and medium level (5-10 instructions/statement). Scripting languages like Tcl tend to be weakly typed and very high level (100-1000 instructions/statement).

Embeddable Runtime.

Mono Virtual Machine:

- Embeddable in C/C++ applications.
- ~4Mb footprint for basic setup (uncompressed).
- Allows C code to call managed code.
- Allows managed code to call into main application.

Fast:

- JIT engine provides the speed.
- Choice of languages.

New Found Users.

Mono is being adopted by game developers.

- For extending their own games.

Unity: 3D Game Development Made Easy

- C/C++ Core
- Mono for high-level operations.
- Multiple-languages: JavaScript, Boo and Mono.
- A language for each task.
- AI, behaviors implemented in high-level languages.

Demo
Unity-based games.

Second Life.

Virtual Reality World.

- Currently using their own LSL-scripting language.
- LSL is a C-like language.
- Speed not very good.
- 3000 computers in August, growing at 300 machines/month.
- 12,000 distinct scripts, 3 million lines of script code (user code)

Mono:

- Gives 50-150x performance increase in scripts.
- Access to more languages, specialize the AI.
- Programs consume half the memory.

GUI Toolkits

Three GUI Toolkits

Gtk#

- Native for Linux
- A .NET binding for all GNOME APIs
- Mike's presentation.

Windows.Forms:

- Almost there.
- Support for 1.1 on the initial release
- Chris and Rolf presentation.

Cocoa# and Dumbarton

- Framworks for building native OS/X applications

Gtk# and the Desktop

Performance

Mono Runtime: Today.

1st generation: interpreter (2001)

2nd generation: x86 JIT compiler.

3rd generation: cross platform JIT compiler.

4th generation: advanced optimizations.

5th generation: pre-compilation.

Conservative Gargage Collector

- non-compacting, non-moving.

Mono Runtime: Future.

New Optimizations:

- Massimiliano Mantione presentation.
- Zoltan Varga's linear-IL representation.
- Massi's new register allocation.
- New IR

Garbage Collection

- Paolo Molaro's presentation
- Compacting Garbage Collection for Mono.

Tuning Existing Optimizations.

Inline turned on by default.

- It required tuning existing optimizations.
- Inline has a number of side effects.
- Tune optimizations for new default.

Results:

- 6% improvement XMLMark/SAX.
- 21% Fast Fourier Transform benchmark
  - On x86, on x86-64, the difference is minimal.
- 2.82% (amd64), 5% (x86) SciMark improvements.
- 2.5% Mono bootstrap

New Optimizations, Today.

Partial Redundancy Elimination

- Implemented a full SSA-PRE pass
- Improves performance significantly for benchmarks
- Need to tune, lots of opportunities here.
- SSAPRE is not enabled by default, slows down JIT time.

Results:

- 5% XMLMark on x86-64 (no difference on x86)
- 22% SciMark improvement on x86
  - 7% on x86-64
- Mono bootstrap (3% faster on hot-run, 6% slower on cold run)

New Optimizations: Ahead-of-Time

Ahead-of-Time Compilation:

- Pre-compiles code in a single pass before execution.
- Eliminates JIT startup problems.
- mcs hello.cs startup reduced in half (0.25 seconds).
- Allows heavier optimizations to be used
  - As compilation time is not a consideration

Limitations:

- Not enabled by default in Mono 1.2
- Only available in select platforms (x86, x86-64).

New Optimizations: Work in Progress

Updated IR representation:

- Current IR engine works on trees, introduces black boxes
- New IR uses lists, more transparency for register allocator.

New Register Allocator:

- Updated

Backup Slides

Platforms support.

32 bit:

- x86
- SPARC
- S390, IBM
- ARM family
- PowerPC

64 bits:

- x86-64
- s390x, IBM
- Itanium

Contributed ports:

- Alpha, MIPS (not finished).

Operating Systems:

- Linux
- Solaris
- MacOS X
- Windows
- Nokia/Maemo

AOT support:

- Based on ELF
- Shared Libraries
- Position Independent Code
- Only on x86 and x86-64

Mono Licensing.

Licenses chosen to maximize adoption.

Open Source licensed.

- Standalone compilers: GPL
- Mono Class libraries: MIT X11
- Runtime engine: LGPL

Novell retains the copyright

- Dual license for compiler and runtime under other terms.
- Novell relicenses Mono for embedded vendors.

Development of Mono

Mono Development

Development Groups:

- Novell, 18 developers.
- Mainsoft, 8 developers.

External contributors:

- 400+ collaborators over the history of Mono.
- 50+ active on a given month.

Not possible without open source community.

Languages

Popular Free Compilers.

C# 1.0, C# 2.0

- Work on C# 3.0 to start soon.

Java

- IKVM library provides Java compatibility.
- Uses GNU Classpath

Boo

- Explicitly typed, Python-inspired language..

IronPython

- Microsoft's own open source implementation.

Nemerle

Phalanger

- PHP compiler, commercial, recently open sourced.

Visual Basic.NET

New Compiler:

- A new from-scratch effort to implement VB.NET
- New version implements VB.NET 8 (Generics and My support)
- Written in VB.NET
- By Rolf Bjarne.

Old Compiler:

- Based on a very old mcs compiler
- Did not keep up with mcs updates, stalled.
- Not worth investing on it.

GCC CIL Backend.

GCC languages can target CIL

- Developed by ST MicroElectronics.
- Backend runs after GIMPLE phase, before RTL.
- Currently configured as a cross-compiler.

Coverage:

- Today: C and C99 as found on gcc.
- With some GNU extensions, but not everything supported
- (__asm__ is not supported).
- No managed extensions yet.

libc: under discussion (p/invoke or port?)

- Likely a CIL libc, to produce portable C applications.
- Possibly use native libc, with P/Invoke.

Other Compilers Under Development

PHP.NET, maybe phased out by Phalanger:

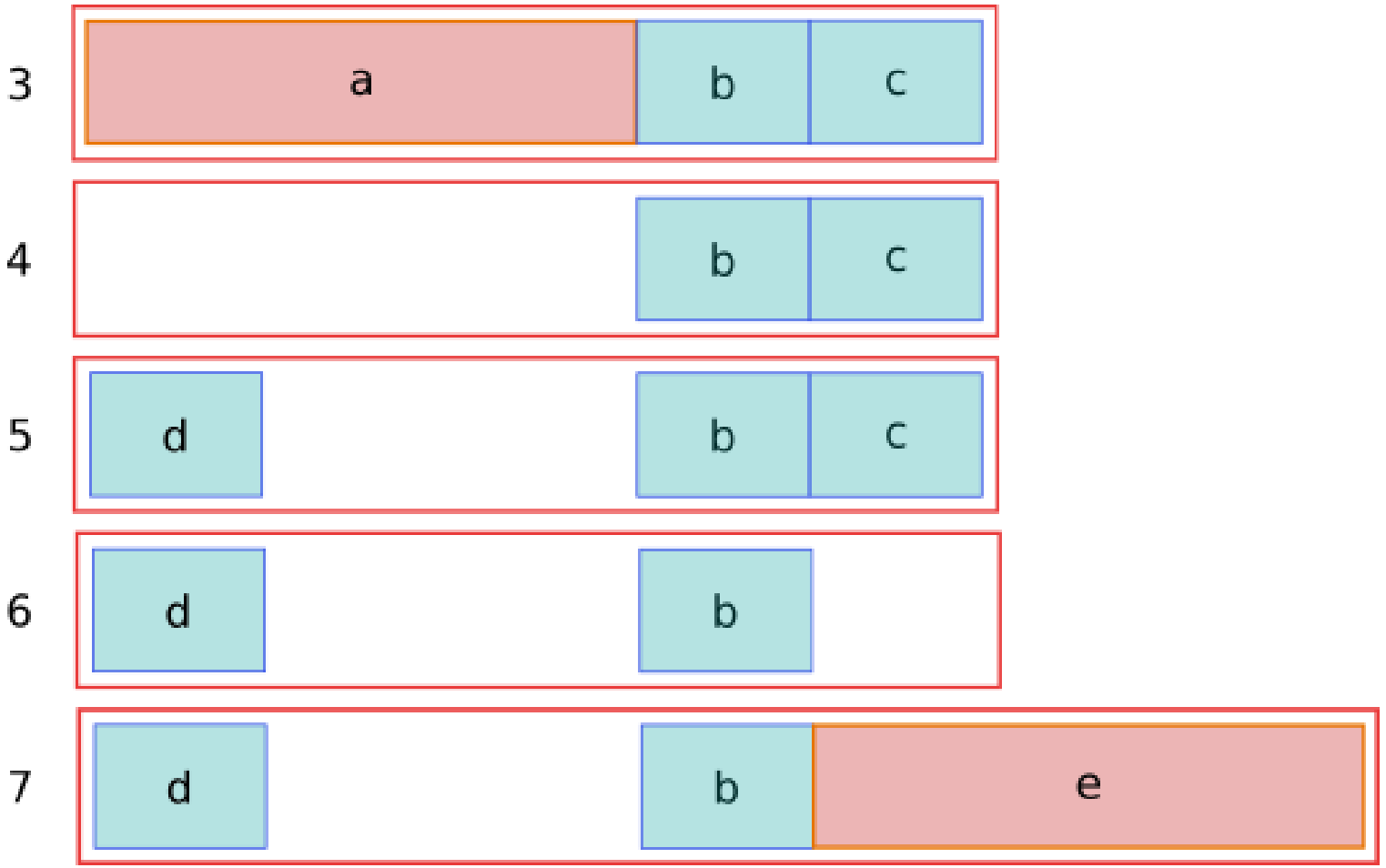
- Goolge Summer of Code (2005)

Ruby.NET:

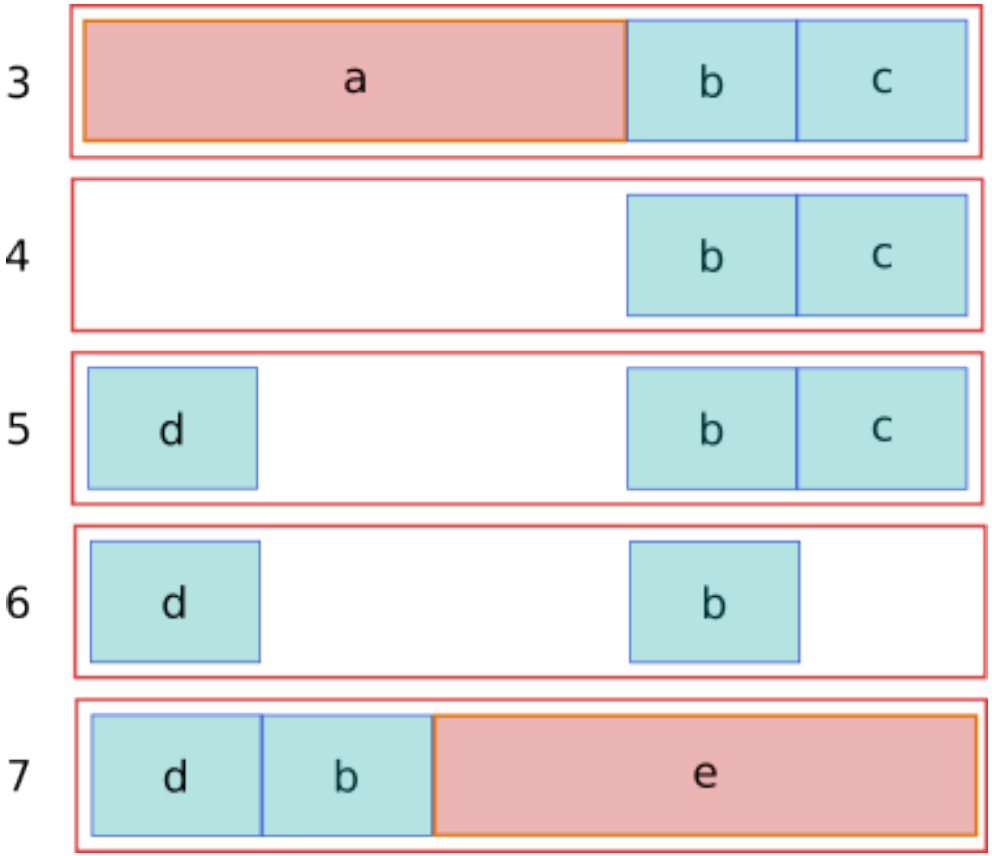
- Queensland's compiler.

Garbage Collector

Compacting Garbage Collector.



Compacting Collector



Compacting Garbage Collector

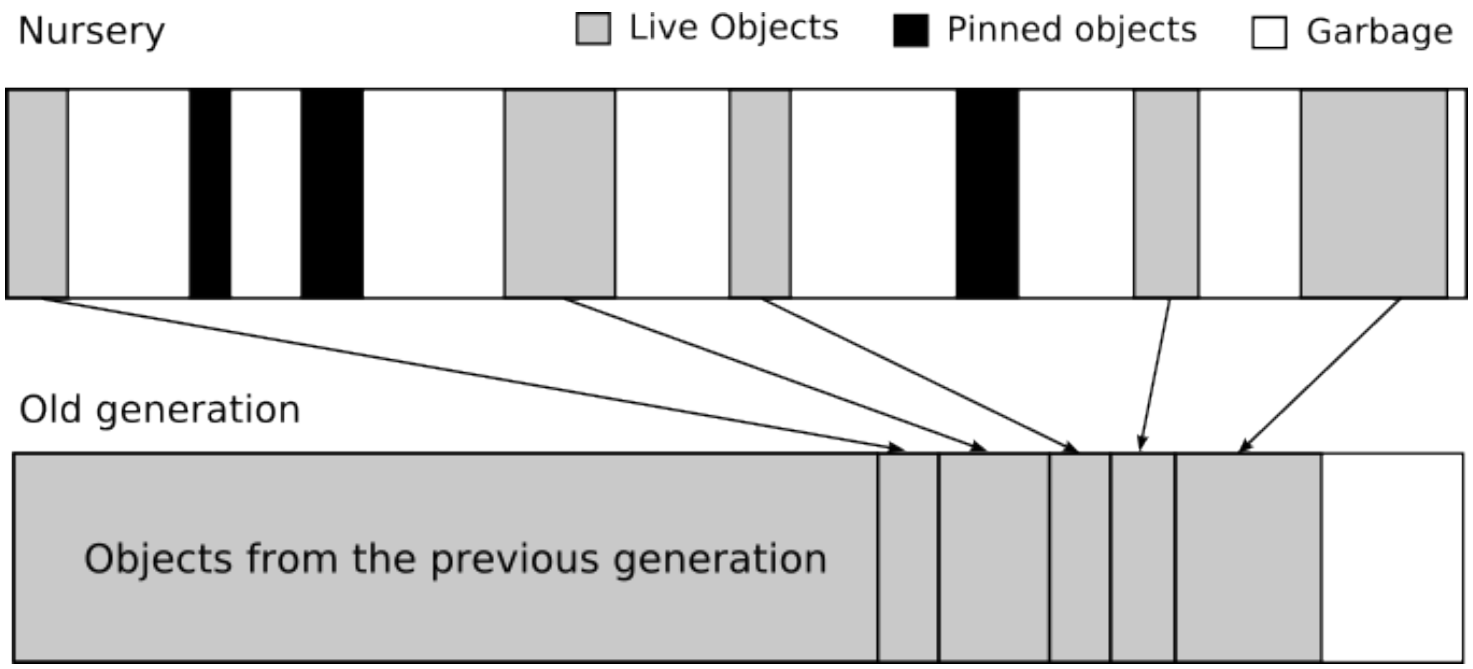
Generational

- Precise
- Compacting

Multi-threaded

- With per-thread nurseries for fast memory allocation.
-

Nursery.


Nursery
Live Objects
Pinned objects
Garbage
Old generation
Objects from the previous generation

Nursery and Pinned Objects.

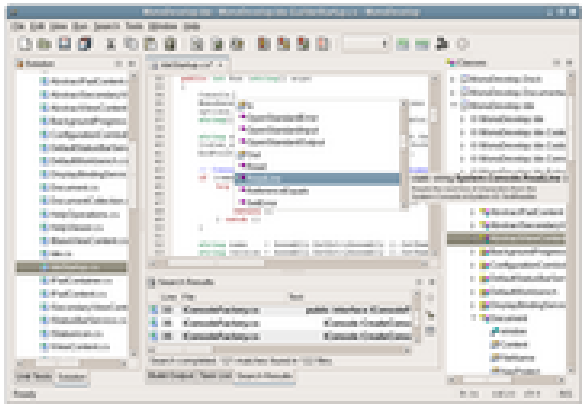
Other.

Developing with Mono.

MonoDevelop

- A GNOME IDE for .Net languages:
  - Based on SharpDevelop.
  - Integrated with Stetic
  - MonoDoc.

MonoDevelop Session with Lluis.



Mono and VisualStudio 2003

Visual Studio plugin allows:

- Test Winforms and ASP.NET apps with Mono on Windows.
- Test with different Mono version

See: Session from Francisco.