



# GT.M+PIP – Past, Present & Future

K.S. Bhaskar  
SVP, Fidelity National Information Services  
[ks.bhaskar@fnis.com](mailto:ks.bhaskar@fnis.com)  
+1 (610) 578-4265

# Themes

- Past
- Present
- Future
  
- Positioning
- Features
- Business model
  
- Community and Ecosystem

# GT.M Today

- The most widely used MUMPS in banking and finance
  - Primarily because of FIS Profile
  - Live since 2005 at the largest real-time core processing system in daily production use at any bank anywhere in the world
    - Processing volumes have grown 50% since then
    - Implementations underway for even larger systems
    - Completed successful benchmarks 1.5x largest system on x86 Linux platform; 3x largest system on proprietary UNIX system and partial benchmarks of systems twice that size
    - Largest single database file created in a benchmark: 2TB
  - Increasingly used in healthcare
  - Widely used
- Loyal user base

# GT.M – widely used



# Market Position

- Accepted as a bona fide MUMPS implementation
  - But way behind the other MUMPS implementation in visibility except in VistA world
  - Within VistA world, generally accepted as a MUMPS platform on par with the other MUMPS – more widely outside the VA for VistA than Caché
  - Some increased visibility anticipated over next year or so
- FOSS business model
  - Limited business success selling support – problem is software is too robust
  - Not so bad that we will stop doing, not so good that management embraces it
  - Widely used as FOSS packages go – usually in top 1-2% of projects at Source Forge by activity

# Life is one long experiment

- 32- vs. 64-bit GT.M on x86 Linux



# FIS' GT.M investment strategy

- Enhancements tied to revenue
  - Needed for FIS Profile
  - Funded by customers
  - Needed to retain customers / non-FOSS opportunities
  - New FOSS opportunities
- Bug fixes
  - High impact bugs – wherever they are reported
  - Low impact bugs – paying customers get priority

# Types of enhancements

- Ports
- Performance
- Functionality



# Ports

- V5.3-000 – support for HP-UX and Linux on Itanium (Q4, 2007)
- V5.3-001, -002 – support for 64-bit processes on AIX, SPARC Solaris, x86\_64 Linux
  - X86 Linux is only platform on which 32- and 64-bit processes will continue to exist
- Future [***usual caveats about the future***]
  - z/OS (anticipated delivery Q1 2009)
  - Cygwin (experimental; 32-bits)



# Performance

- Workloads
  - Need a workload to improve performance
    - Since Profile is handy emphasis has been on TP
  - No obvious bottlenecks identified & performance is not out of line with other MUMPS – other customer benchmarks and VistA
- Memory semaphores on Linux
- Critical section shortening
- MM (field test V5.3-002; fully supported next release)
- Future
  - Compression of replication stream
  - Database triggers (under consideration)

# Functionality

- Recent
  - Multi-site replication
  - Replication without before-image journaling
  - Continue replication in the face of temporary journaling failures
- Future
  - IO to a process
  - Alias variables
- Future direction
  - Logical Multisite eXtension (will probably require TP to be useful)

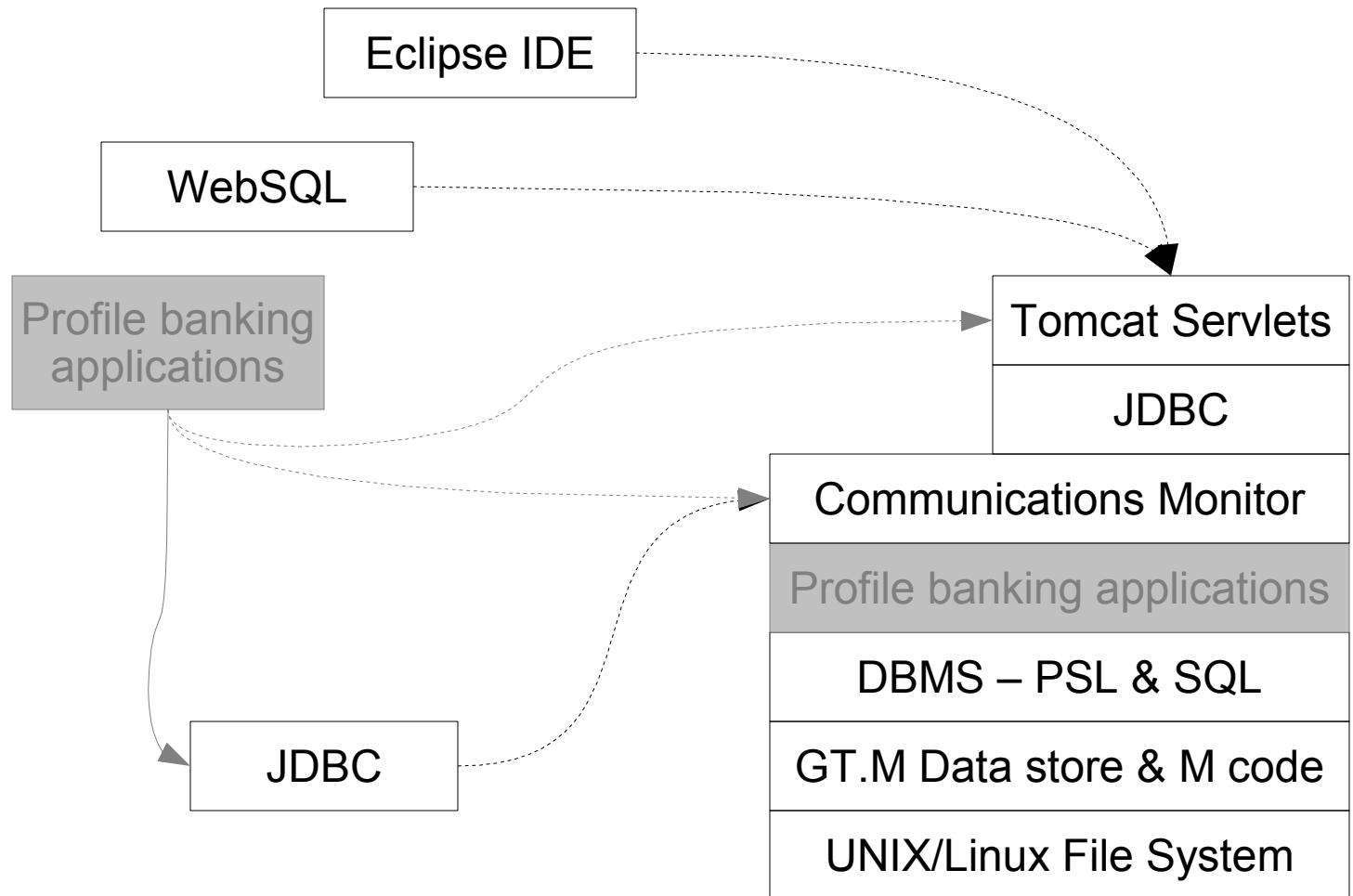
# Life is one long experiment, continued

- 32- vs. 64-bit G.T.M on x86 Linux
- PIP

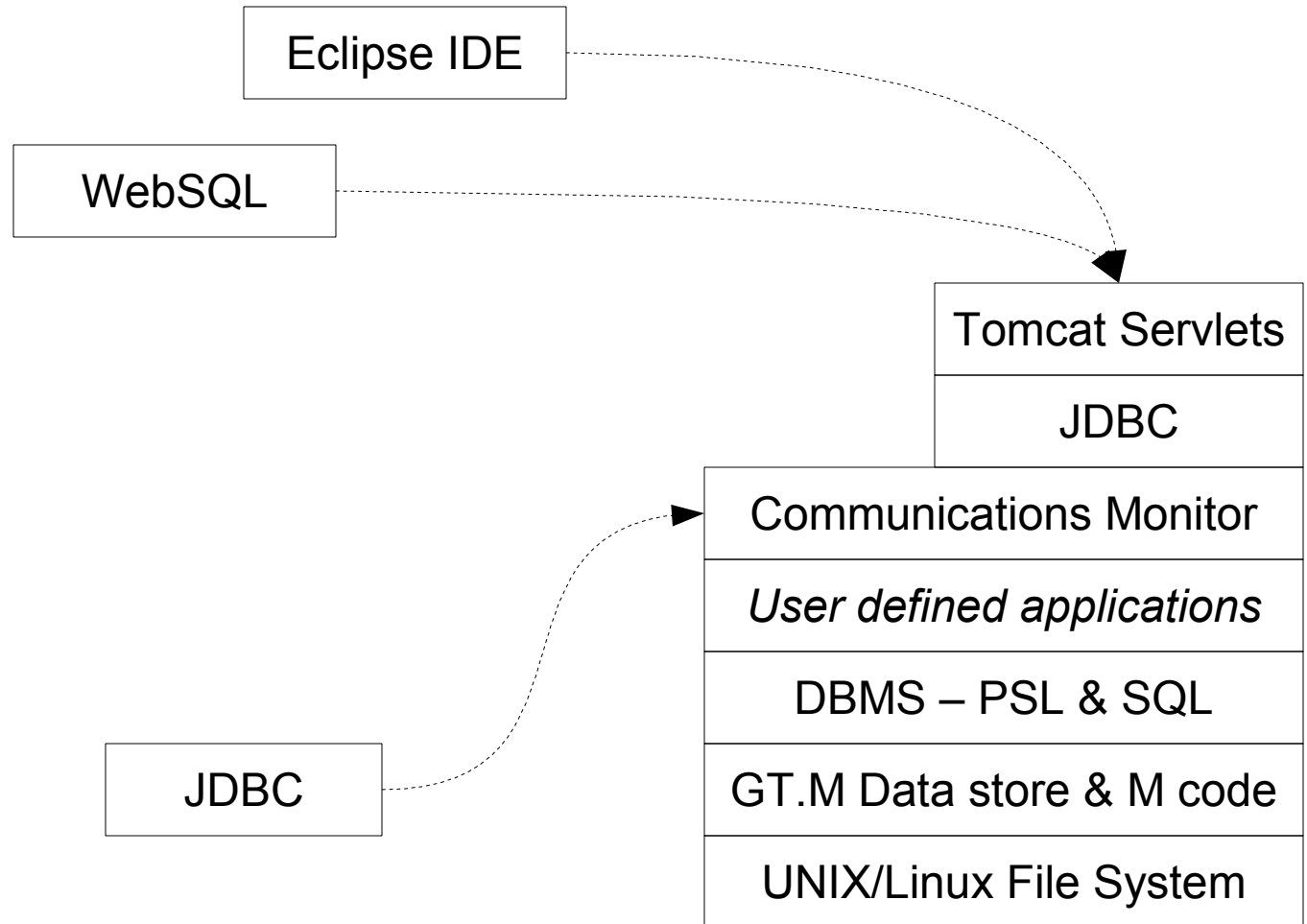
# PIP – Infrastructure from Profile

- SQL
- PSL (Profile Scripting Language)
- Communications monitor
- JDBC driver
- Eclipse IDE
- WebSQL
- *(ODBC driver optionally available, but not FOSS)*

# Profile Architecture



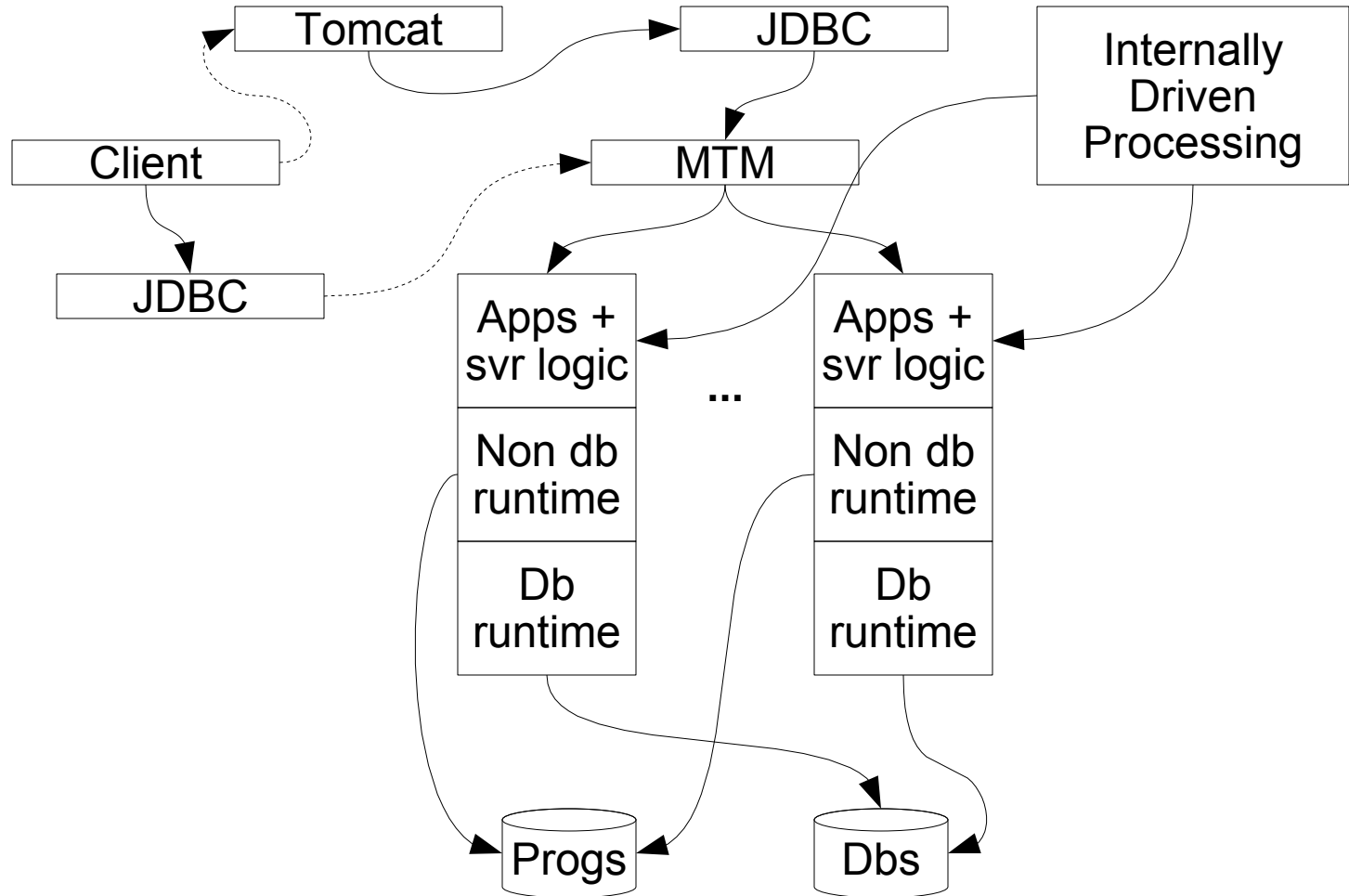
# PIP Architecture



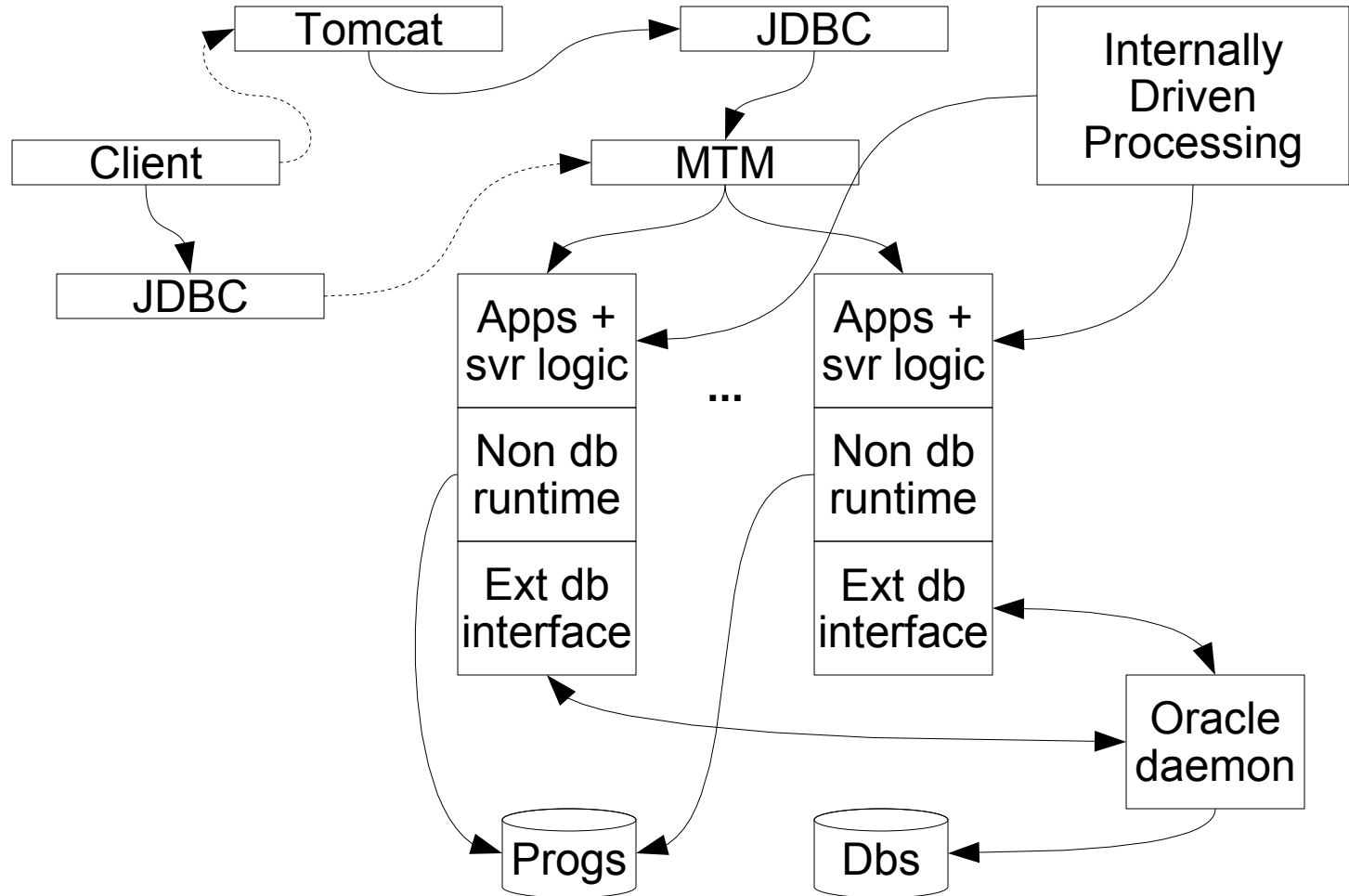
# PIP Technology

- Applications developed in:
  - SQL (or on client via JDBC)
  - PSL (Profile Scripting Language – lightweight objects)
- Compiled into M
  - M acts as embedded virtual machine, like JVM for Java
  - Target choices: M database or off the shelf RDBMS
  - Today GT.M is supported M; Oracle is supported RDBMS
  - Architecture allows for future additional M & RDBMS targets
- M is compiled into native machine code by GT.M
- Operational performance and functionality determined by database engine (e.g., logical multi-site supported with GT.M)

# PIP/GT.M Runtime environment



# PIP/Oracle Runtime environment



# PIP

- PIP v0.1 packaged and released as proof of concept virtual machine, January 2007
- PIP v0.2 expected October / November 2008
  - Cleaned up dependencies (e.g., won't need DbVisualizer to create tables)
  - Support will be available on commercial terms
  - Released as virtual machine and as package
- Future
  - Improved documentation
  - Features as they are released for Profile

# Technology strategy

- GT.M – underlying engine
  - Robustness
  - Performance
  - New features to facilitate operations and PIP
  - For legacy / heritage MUMPS applications
- PIP – application development layer
  - New applications
  - Evolution of MUMPS applications

# Life is one long experiment, the future

- 32- vs. 64-bit GT.M on x86 Linux
- PIP
- Schema-less databases
  - What can we do together as a community?
  - How can we create a viable ecosystem?