Mavimax

Mavimax, Ltd

Enduro/X Middleware Data Sheet

August 2022

EXECUTIVE SUMMARY

In today's business environment of system consolidations, worldwide utilization, and "always-on" availability, enterprises need distributed transaction processing (TP) infrastructure that leverages existing skills, application assets and new technologies, while delivering standards based interoperability for simplified integration.

Mavimax Enduro/X is today's platform of choice for C, C++, Go, Java and Python solutions providing the backbone for the world's leading companies, running some of the largest mission critical data processing systems— wire transfers, ATMs, POSes, telecom, aerospace and defense. The reason for this is that only Mavimax Enduro/X provides the rugged platform required to run these high volume applications across distributed, heterogeneous computing environments; enabling transactions that stretch from customer-facing, business-critical applications to back-office processes, across any system, anywhere in the world.

INTRODUCTION

The world is tending to more and more digitalized every day. The Internet Of Things (IoT) is coming to our lives, and our lives become connected to the cloud computing environments. As the world is thirsty for the new integrated cloud solutions, the vendors have a hard time for creating large scale massive applications that would integrate with different kind of software packages that could be deployed in high availability (HA) and high performance (HP) clusters. Thus many companies create their frameworks by them selves, which initially might look a good solution, but with time the demands grows as well as the different programming languages are becoming involved into project. This growing from zero causes a lot of problems when then project is grown – because of the non scalable fundamentals have been employed. This is leading to the rapid increase to the total cost of the ownership (TCO), the maintenance of the system becomes very hard, new functionality implementation also requires lot more of investments, due to fact that in-house made, non scalable interfaces are used.

Thus industry is requiring and good platform which would be future proof – scalable, stable and expendable. Many technologies are emerging to this area, but there are only few or none packages that would cover the all demands on high integration, high availability and high performance cloud backbone solution.

PROBLEM

Currently in marked there are middlewares, for example ZeroMQ, RabbitMQ, etc. But these are just messaging systems. If you start a new cloud business are extend your existing one with these emiddlewares, then the principles how your business services communicates in between are up to you. The management of the services and processes are not covered by these software packages. The process run-times, configuration and logging in different languages and environment are done differently. This makes systems to appear as non homogeneous, chaotic.

The markets now are tending towards the HTTP REST API usage over the micro services, as these are standards based and available on different platforms. However, for building open systems with high throughput these technologies are not suitable because of the very high overhead for the message processing.

SOLUTION

To address the requirements above, Mavimax is providing Enduro/X middleware product. Which is by comparing to other competitors more sophisticated, more scalable and modern middleware product. Enduro/X provides:

- Open system model.
- Application server.
- Strong failure recovery and self healing.
- High performance.
- Distributed Inter Process Communications (IPC) middleware.
- Data Dependent Routing.
- Out the box service load balancing.
- Unified API for multiple programming languages.
- Distributed transaction processing and transaction manager.
- Client process monitoring.
- Persistent messaging.
- Unsolicited messaging.
- Event broker.
- SOA Service Caching – Enduro/X Smart Cache.
- Connectivity to Web resources and services.
- Generic IoT and industry specific connectivity.
- Common Configuration Framework.
- Common Logging Framework.

- Latent Command Framework.
- Code Generators.

Enduro/X is built on a service-oriented architecture that utilizes a powerful messaging and transaction processing engine to dramatically simplify enterprise distributed computing. Leveraging powerful, yet simple-to-use APIs and integrated services that abstract the complexities of distributed transaction processing, Mavimax Enduro/X improves time-to-value for your most ambitious and complex distributed projects. Enduro/X supports dynamic system reconfiguration with out service outage, due to fact that most of the services are running in replicated services mode. Mavimax Enduro/X ensures the highest reliability and performance for your most demanding mission-critical applications. By dynamically load balancing service requests among all the available systems, Enduro/X enables applications to handle requests in parallel and process multiple transactions simultaneously on different, distributed cluster nodes. For flexibility in processing or deferring transactions, Enduro/X provides transaction queuing; allowing distributed applications to work together in an asynchronous, “connection-less” fashion.

Enduro/X is built for Mission-Critical A highly reliable, highly available, high-performance system is realized for further improving business continuity. The core of the Enduro/X encapsulates all of these properties. Depending on OS support Enduro/X utilizes Posix Kernel or System V message queues which are proven answer to the today demands for multiplexed systems. Enduro/X is designed to keep track of the all of the processes and monitor their life cycle. The system is intended to heal it self in case of unexpected software or hardware malfunctions.

Open system model

The Enduro/X is build on open system principles. Meaning that target developed system is composed of separate executable processes and implements best Unix practices – “*Make each program do one thing well*”. This principle supports very large application development, i.e. building application of small blocks of well developed and tested units. The block for application building can be XATMI server processes, XATMI client processes and script. With help of the Mavimax Enduro/X all these blocks are combined and orchestrated into single application unit.

Application server

One of the Enduro/X functions is to server as application server. The system contains configuration files for list of the processes that needs to be started and monitored. Enduro/X use these descriptors to boot the services, monitor their health and do the graceful application shutdown when needed. The application server basically is operating system local process container which manages the processes life cycle. This is one of the features to answer the question on the homogeneous infrastructure building. The unified interface to application server as the local cluster node, makes cloud infrastructure building more streamline and standard.

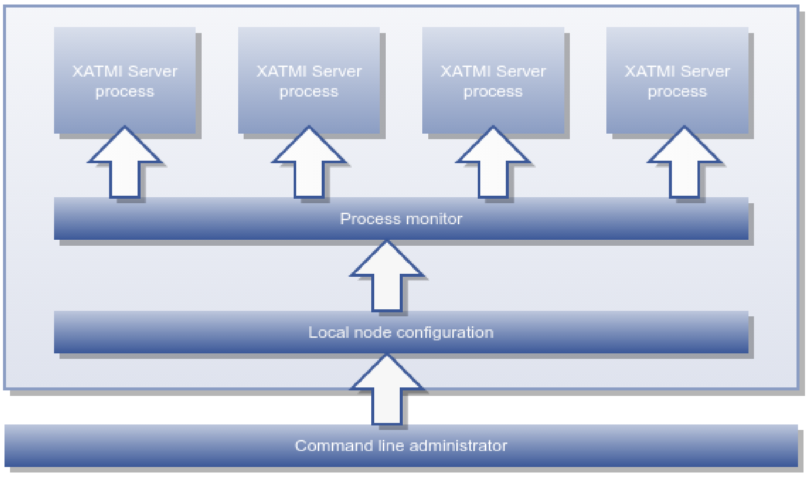

Figure 1: Enduro/X process containers

Another useful feature of the XATMI platform is that platform supports **real-time application patching**. This means that application components – stateless services can be replaced with new binary with out service interruption. As the services can be started in multiple copies, the loading of new executable is possible by doing one by one restart.

Strong failure recovery and self healing

The Enduro/X Daemon on each machine monitors the state of all servers and can automatically restart failed servers. It can also detect hung servers and kill/restart them as required. As the cluster bridge processes are part of the servers booted by Enduro/X Daemon process, thus it monitors a cluster links, which can be restarted if crashed or not connected. Enduro/X Daemon can be monitored too by special process which can detect it's failures and restart it. After the restart daemon will learn about the system and after period of time will start acting as a normal. Thus there are no single points of failure. Any transactions that are affected by a server or machine failure and that have not completed the prepare phase are rolled back. Transactions that have completed the prepare phase but not the commit phase will be committed as part of the Enduro/X boot sequence.

High performance

Enduro/X is built with performance in mind. On local server for IPC it uses Operating System's kernel queues. For networked cluster, a high performance multi-threaded bridge process is used. The figure 2. Enduro/X benchmark on commodity hardware. Even so Enduro/X middleware is performing quite well under the sub-millisecond response times. In particular example one request and reply pair was only 13 microseconds on Linux, x86 platform, which is quite impressive.

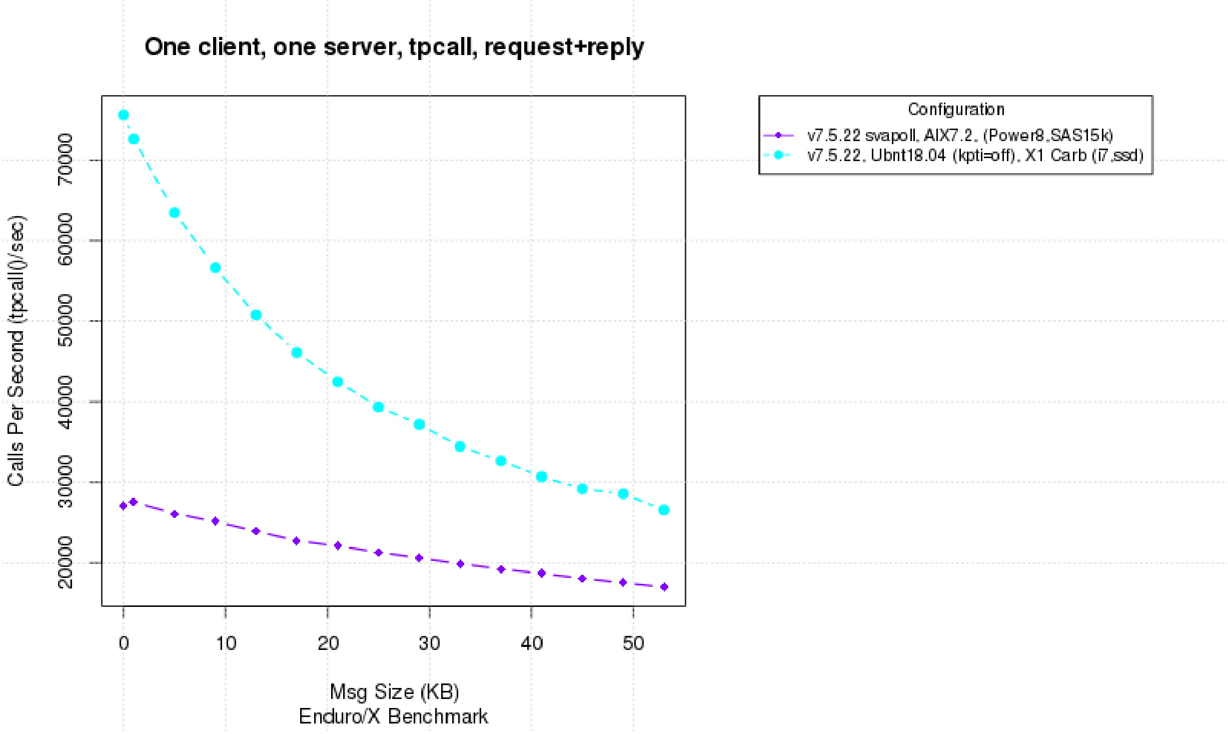

Figure 2: Benchmarks on low end hardware

Distributed IPC

Middleware is core functionality of Enduro/X. The middleware follows the XATMI patter and basically is call system which is build on top of the queues. The call API does the abstraction for local process calls and remote cluster node server calls. From developer perspective there is no difference between these. All the developer cares about is the service name that needs to be invoked. This greatly simplifies the application development and does not require any special works to be done for application deployment in cluster. The developer and system architects can care about services and their logical work and the physical configuration is left to the deployment time and can be managed by integrators or system administrators.

Enduro/X offers different IPC patters: synchronous/asynchronous request-response calls, conversational IPC with invoked service (multiple bi-directional message exchange with in single service invocation) and publish-subscribe event posting.

Distributed transaction processing

Enduro/X provides transaction control based on two-phase commitment to assure data integrity in a distributed computing environment. This enables data updating across multiple resources and global transaction control across multiple servers. The transaction stages on resources are logged to persistent disk storage. Thus full or partial recovery can be automatically performed after a system failure. When the Enduro/X system is linked with a database management system (DBMS) through the XA interface of the DTP model defined by The Open Group (formerly X/Open), recovery instructions are sent to the DBMS when the application is recovered.

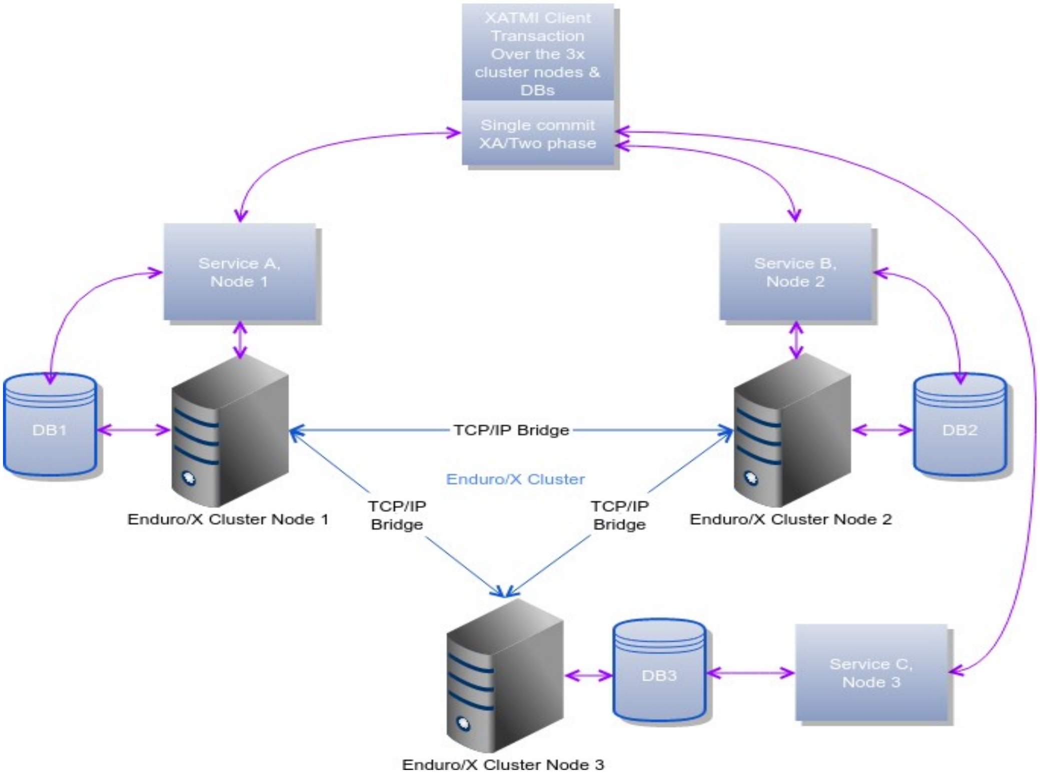

Figure 3: Enduro/X cluster configuration with 3 nodes and single commit

Distributed transaction functions across all supported programming languages: C/C++, Go, Java and Python. Java JDBC drivers enable wide variety of supported databases which can be involved in the solution.

Out the box service load balancing

Enduro/X is built on kernel based in memory POSIX queues. Each service that is provided by XATMI server executable process is available for client service calls. The service names that are provided by servers can be duplicate. In that case the load balance is automatically activated. The service calls now are dispatched in load balance manner to the server processes which share the same service name. These works with remote clustered services too. If service is available on cluster machine, the calls can balanced between local servers and remote servers. For certain platforms like Linux and BSD, shared services are load balanced in “One queue – multiple servers” manner, which means that any free first XATMI server process will consume the incoming workload. For other platforms round-robin approach is used.

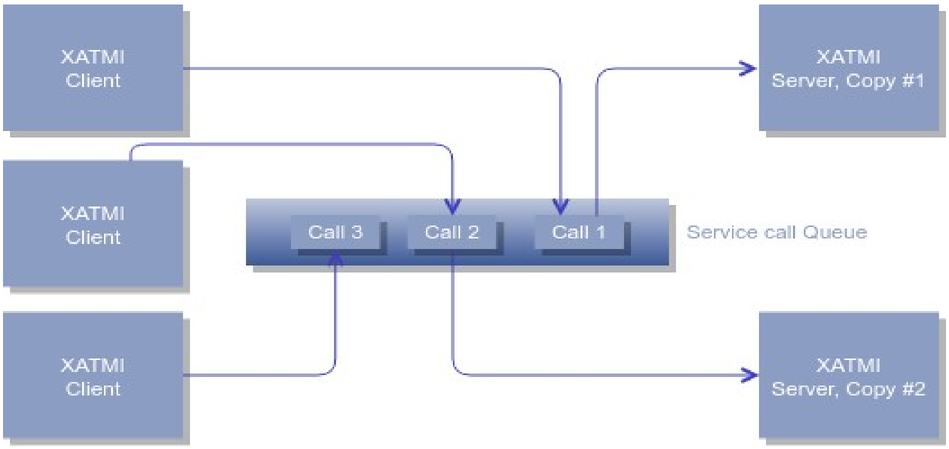

Figure 4: Service call queue - on queue multiple servers, call 3 enqueued

Data Dependent Routing

Data Dependent Routing (DDR) allows to shard single application into multiple logical clusters by means of administrative configuration change. DDR functionality allows to scale the application across several resources with out requiring to perform any changes to the code. This might be useful for example in scenarios where for security or performance reasons different physical databases shall be used. To enable routing, application shall have some data property in service call buffers that can be used to map the particular call to routing group. When caller makes a service call (see figure 5.), routing details are checked and evaluated after which client process decides to which group/shard finally to send the request. Routing lookup is quick lock-less operation. The services and routing expressions are registered in the Enduro/X configuration file. Routing data is cached in shared memory and routing data may be reloaded in real-time.

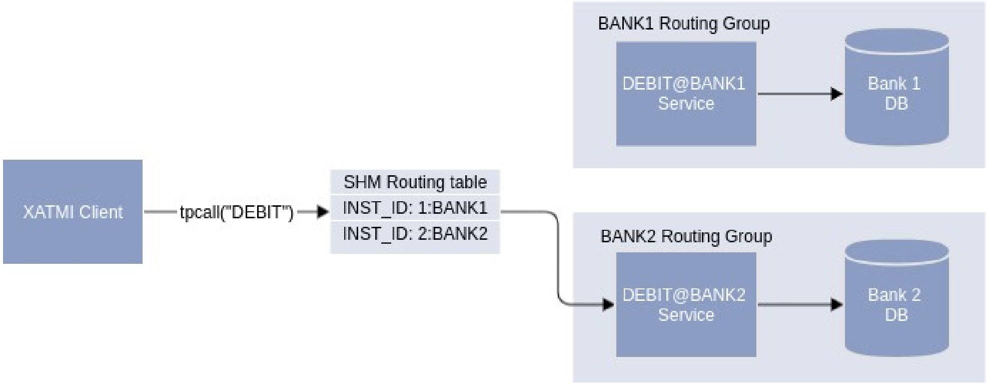

Figure 5: Data Dependent Routing call principle

Unified API for multiple programming languages

Enduro/X implements extended XATMI standard API calls. The API commands includes different set of functions for doing service invocations, service management, message forwarding, distributed transaction management, multi contexting. In total about 70 function calls are available for XATMI. Also Enduro/X provides different set of buffer management APIs. Buffers are as a transport format of IPC calls between services. Buffers are backbone of your application. Supported buffer formats are following:

- **Unified Buffer Format (UBF)** is a protocol buffer which provides basically a typed fields that contains numbers, chars, strings and raw data. Each field can contain array of elements. The field tables are predefined and compiled as a numbers for fast data access. The buffer fields are indexed by binary search. And data is located in single allocated memory block, which means that easy buffer copy, store and restore is possible. This is in contrast for example with JSON APIs, where the indexing is done by name and usually the parsed data is located into segmented memory, thus buffer manipulations are somewhat limited. UBF buffer provides emulation of Oracle Tuxedo FML, FML32 API, and is highly compatible. Including support for recursive buffers, pointers to XATMI buffers and VIEW buffer fields. UBF buffer management consists of about 120 API calls. These are provided to Enduro/X supported languages. For Go and Java programming language extensions are provided for easy data marshaling and unmarshaling from/to structures. In Python programming language, all data buffers are projected to dictionaries.
- **VIEW** buffer allows programmers to send complete objects from on process to another in cross platform way. Objects are described in view file from which corresponding structures are generated for target programming language. At the time of the call, the structures/objects are serialized, transported over the IPC and deserialized in target process automatically. View buffer can be optionally projected to UBF/FML buffer. Enduro/X offers unique API for accessing view fields dynamically from the code. With this functionality out of the box scripting link interfaces can be enabled for VIEW buffers.
- **STRING** buffer is basic NUL terminated string. The manipulations are standard one as in C with allocated string. Buffer read and write functions are available for Go and Java. Dictionary is used for Python
- **CARRAY** is raw data buffer format. The manipulation are the standard as in C byte arrays. Buffer read and write functions are available for Go and Java. Dictionary is used for Python
- **JSON** buffer also is supported. Enduro/X includes functions for automatic buffer conversion to and from UBF buffer. This can be suitable for using internal UBF buffer as normal form and using converted JSON buffers for external connectivity.

Client process monitoring

Enduro/X package includes client process monitor. This module allows system architects to connect arbitrary processes to Enduro/X application suite. This module basically offers automatic process management, like auto start, restart when process died for some reason and administrative command line tools, like listing status, starting and stopping particular processes.

Persistent messaging

Enduro/X middleware offers persistent message queue API. Queues provide time-independent communications fro application that do not need to communicate interactively Queued message are stored to disk for guaranteed message delivery. Queues can be automatic – messages are automatically delivered to configured target service. Or queues can be manual, processes can poll the queue names and receive messages manually. Queues can be FIFO or LIFO ordered. Persistent queue sub-system of the Enduro/X are transactional and transaction can participate in global transaction among the other resource transaction, e.g. database.

Unsolicited messaging

Enduro/X middleware offers API for delivering unsolicited messages to one or more client processes. Two classes of notifications are available: “**notify**” process which is actually is in synchronous service invocation (for example report back the progress) or “broadcast” which allows to send unsolicited messages to client processes matched by specific parameters or wild-card.

Event broker

The event subsystem within Enduro/X provides support for brokered events. Brokered events allow application to subscribe to events of interest and when another application posts an event, all applications subscribed to that event receive it. This allows applications to use an event driven model instead of the more typical request/response model. Evening is done in cluster wide scope. Thus service on one machine can subscribe to events which are produced on another machine.

SOA Service Cache

Enduro/X Smart Cache feature provides service call caching. Administrators or system vendor can provide additional configuration for Enduro/X applications which enables certain service caching. Configuration includes different rule definitions – at which conditions data must be cached, what data needs to be cached, how long data stays in cache and how to perform any cross invalidates. Enduro/X Smart Cache provides cluster operations, were data cached on one cluster node is immediately sent to other cluster nodes via Event Broker. Other cluster nodes may subscribe to certain events (like data put or delete). Administrative tooling is provided for cache maintenance in runtime (view and delete cached data).

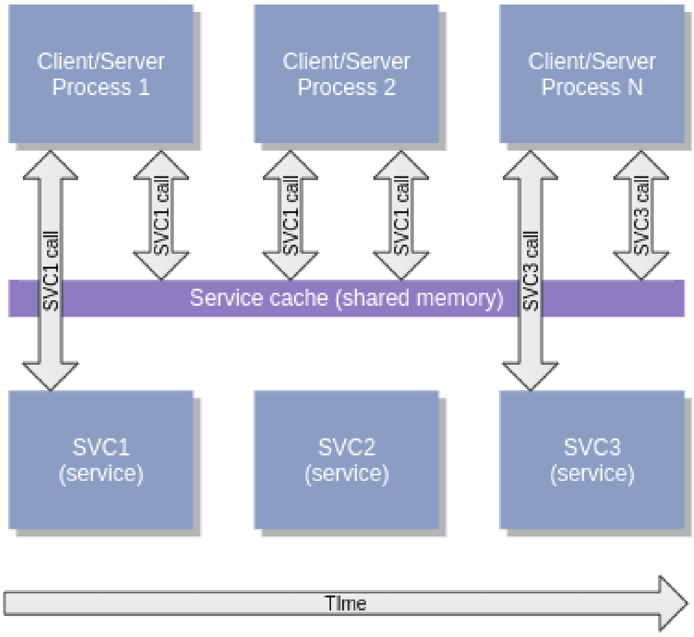

Figure 6: Service cache working principles

Connectivity to Web resources and services

As the we resources and services become more and more mainstream for computing, Enduro/X provides module for mapping the external HTTP/REST service as the XATMI service, thus there is no need for special third party libraries to access Web Services from C/C++ or other Enduro/X supported languages, see fig. 7.

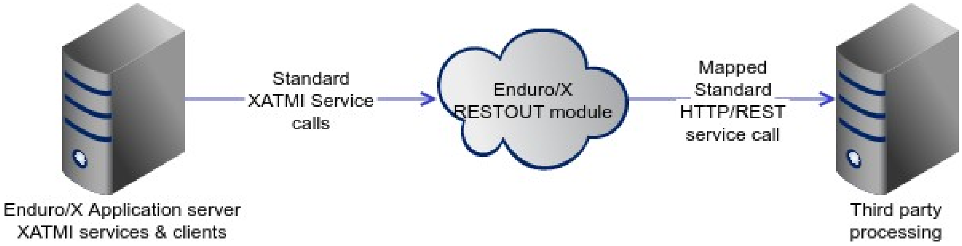
Standard XATMI Service calls
Enduro/X RESTOUT module
Mapped Standard HTTP/REST service call
Enduro/X Application server XATMI services & clients
Third party processing

Figure 7: Enduro/X accessing third party HTTP/REST Web Services

Enduro/X also provides module for exporting existing XATMI server to HTTP/REST, see fig. 8. Thus existing services can be easily mapped to Web Service and third party integration can be done with relative simplicity.

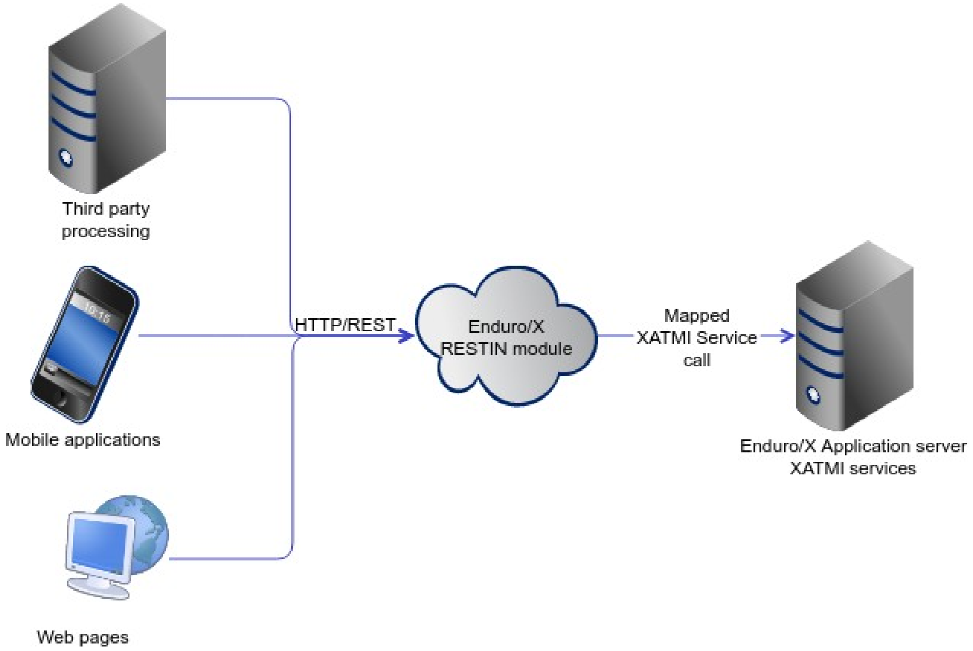

Figure 8: Exposing Enduro/X XATMI services as HTTP/REST web services

Following buffer formats for HTTP/REST In-Out data exchange:

- UBF/FML.
- VIEW.
- STRING.
- RAW data.
- JSON.

Generic IoT and industry specific connectivity

As the demand for smart systems grows, Enduro/X supports low level TCP/IP connectivity package. Which brings the abstraction of “socket” to the next level. Now raw connections are established and managed at XATMI level. To send data to network it is just a matter of usual XATMI service call. The same way if data is received from network, XATMI service is invoked with data buffer received from network. Enduro/X TCP gateway supports client and server roles. The connections are enumerated so that applications can reference particular connection at processing stages.

Package supports different connectivity pasterns, like asynchronous messaging and synchronous messaging over persistent connections. Or request/reply pattern over one connection for one request. Multiple connections can be added to application server, and for example IoT devices can be mapped as services in Enduro/X application which later does the TCP connectivity to actual device, see fig. 9.

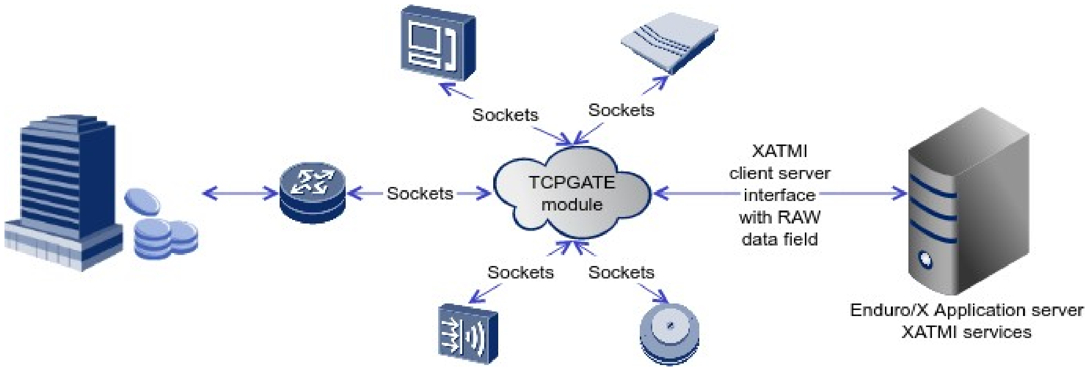
Sockets
Sockets
Sockets
TCPGATE module
XATMI client server interface with RAW data field
Sockets
Sockets
Enduro/X Application server XATMI services

Figure 9: Generic TCP connectivity

Different data packet framing models are supported, including various packet length tag formats and start/end ASCII tokens .e.g. STX/ETX.

Basically Enduro/X is a perfect middleware for IoT processing.

Common Configuration Framework

When it comes to multi language system programming, significant role has a framework. More it is homogeneous the better quality solution we get. Thus one of the things that large scale application is needed is a configuration management. Enduro/X have a solution for this – Common Configuration framework. Enduor/X provides a XATMI server which can return INI file section values in key/value format. The values are requested with help of standard XATMI service call. The underlaying framework supports multiple INI files and merge of the sub-sections for returning result set.

Enduro/X allows to configure application and parametrize Enduro/X settings at XATMI server or client process levels with help of the tagging. The same tags can be used for user specific application parameter reading.

All these features gives very high flexibility for building highly complex and highly scaled applications.

Common Logging Framework

Another significant requirement for high quality software is a having a good debug logging framework. Log files are must have feature for development process and for long term customer support. Thus Enduro/X exposes it’s internal logging framework accessible from XATMI level API. The logger includes following features:

- Different log levels.
- Log buffering.
- Different logger configuration for application threads.

- Per binary external configuration.
- API for logger configuration.
- “Request log-file” feature allows to temporarily redirect all the logging (user and Enduro/X) to separate log file. This is useful if for each user session or transaction separate log file is created.

So by using these logging APIs, it is possible to get unified log file format for all of the programming languages involved in project.

Latent Command Framework

To support “always-on” nature of the applications which are build on Enduro/X basis, middleware provides administrative interface for changing the process logging settings on the fly with out reboot. This administrative interface also allows to perform log rotation with out any loss of logging data. If applications are using Enduro/X Common Logging Framework, then these properties will be automatically enabled in developed applications. Latent Command Framework is extensible via Enduro/X plugin interface, which gives API for application developers to create additional real time commands (such as configuration reload at request, etc.), that may be executed by any Enduro/X related binary in near real time.

Code Generators

Enduro/X ships with code generation framework and initial templates with which developer can simply generate code for basic XATMI client and servers for C/C++, Go, Java and Python languages as well as UBF table definition files. The code generation gives user ability to choose different settings in interactive mode before emitting out the actual result. For example user can configure the new process name, service name, read the common configuration section and use common logger.

The code generation framework uses Enduro/X internal scripting language named PlatformScript, it is very similar to JavaScript, thus developers should not have problems. Typically when developing large systems, there are some patterns when architects select. These patters can be developed as templates and later will help developers to quickly start the work on the system.

If migrating from systems like Oracle Tuxedo with existing code base, then generators will help, just need to develop the code generator templates.

This concept of code generation will help drastically to reduce development costs, because now routine tasks can be done in few seconds instead of hours.

CONCLUSIONS

Enduro/X is the industry’s #1 open source alternative for distributed transaction processing platform. It provides mainframe-class reliability, performance, and scalability, all on open, distributed systems for applications written in C/C++, Go, Java and Python. Enduro/X is the premier platform for building mission-critical applications and re-hosting mainframe applications on open systems and commodity-based grid infrastructure. Mavimax Enduro/X is cost-effective, reliable, and extremely scalable. As companies seek to lower their total cost of ownership, adopt more standards-oriented platforms, and leverage existing IT assets while not compromising on the mission critical nature of their business applications, Mavimax Enduro/X emerges as the platform of choice.

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.