



# Kondor+ Monitor Station

---

## Risk Solution for export/import on Demand

11<sup>th</sup> November 2012

By Antonio Di Sabatino (Senior Front Developer)  
(draft: **delivery at end of March 2013**)



# Agenda

- Objectives and features
- Technologies and Architecture
- Workflow and MetaData
- Xml Transformation and Xslt Engine
- Import Handler as Xslt Gateway
- Export to IBM MQ
- Batch command Tools

# Objectives main features

- Kondor+ Mvts real time blotter monitoring and filtering
- Kondor+ Mvts database storage and reporting
- Export to Import Workflow
- Export on Demand for single static data or deals
- Massive Export on Demand by Table Name
- Import on Demand apply the Transformation Engines
- Massive Import on Demand
- Publish Kplusfeed on IBM MQ and listen for ACK

# Monitor Server

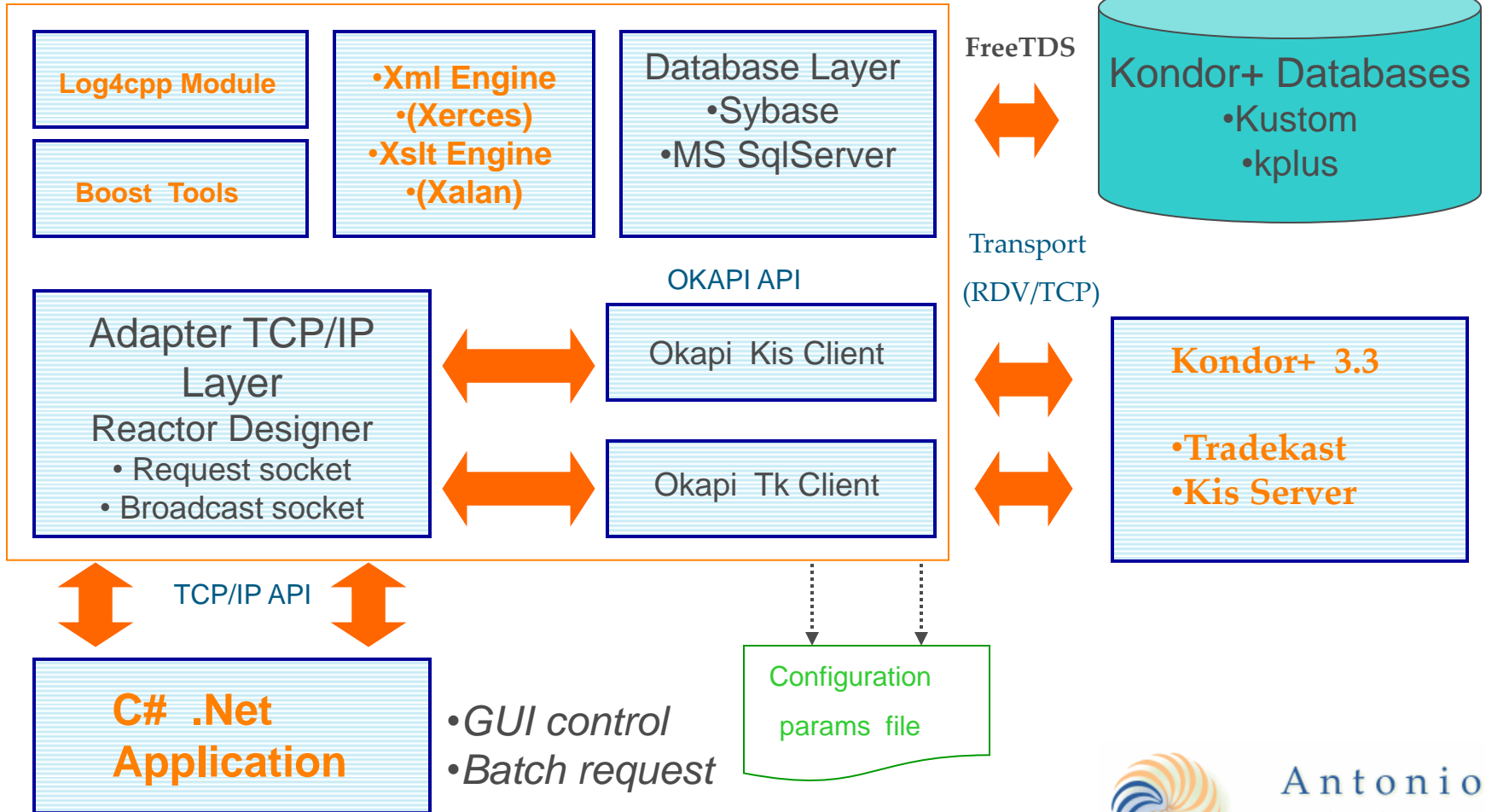
- The Monitor Server is a backend server-side process developed in C++ OOD using the Standard STL, Boost Libraries.
- The Monitor Server will connect to Kondor+ Kis process for importing workflow
- The Monitor Server will connect to Kondor+ Tradekast process for exporting workflow
- The Monitor Server will bind two Tcp Sockets: one is the request socket and the other one is the broadcast socket

# Messages Handlers

- The Monitor Server will manage the requests across a chain of handlers
- Each handlers will be linked to specific *Action* and it will be responsible to manage that specific request
- Each handler will lock the internal resources for the duration of the request
- The Monitor Server is able to manage multi-client connections and multi-different action

# Client Server Architecture

Monitor Server (C++)

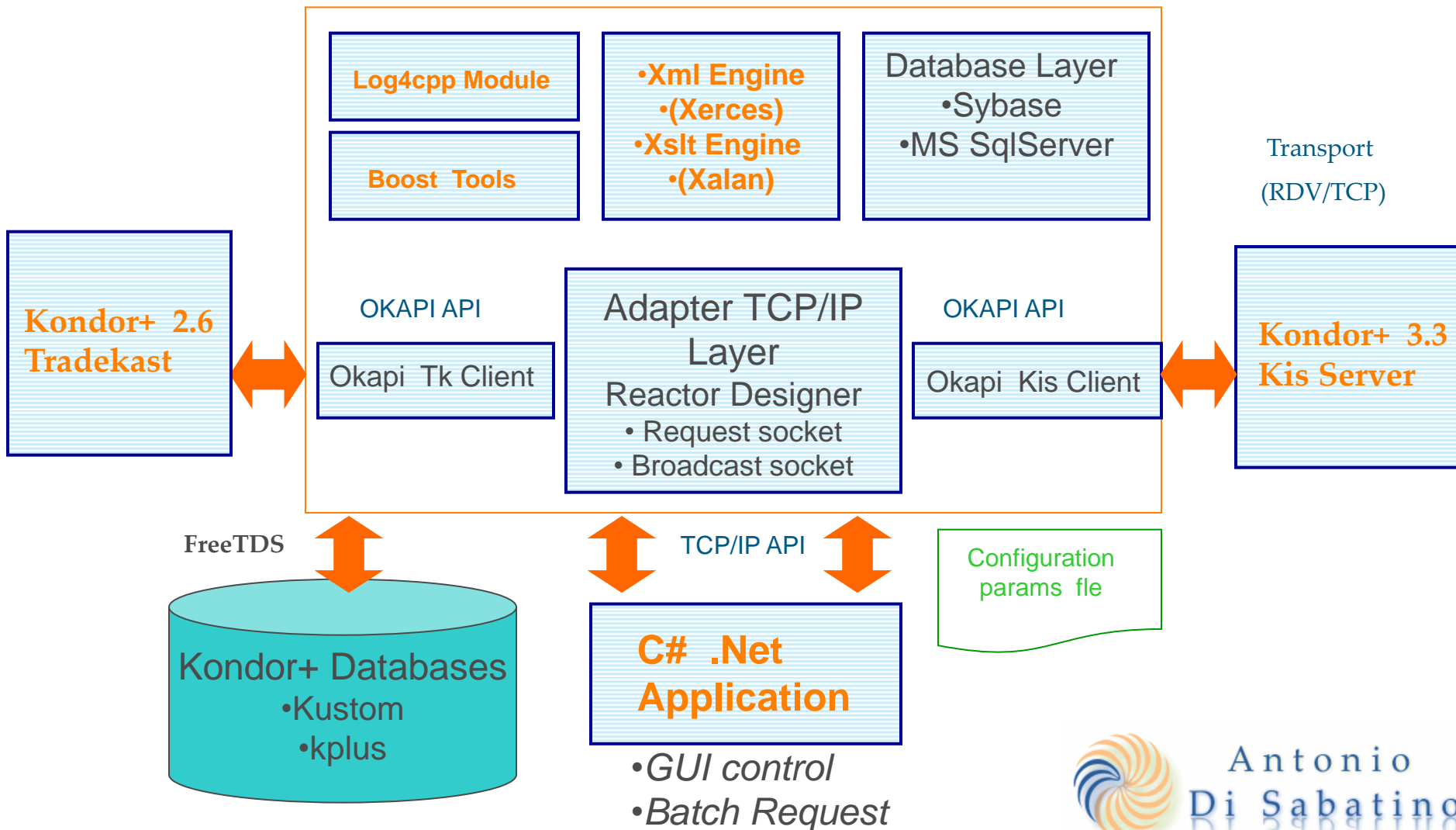


## Export to Import Workflow

- The monitor server implements the Export to Import handler.
- The purpose is to synchronize two Kondor+ different versions one way: (master/slave, production/test, production/DRP).
- Using this feature is possible to input into a central Kondor+ system all the trades coming from several Kondor+ peripherals systems (aggregation)
- Reporting aggregation, testing and monitoring

# Export to Import Architecture

Monitor Server (C++)





# Technology Notes

- Software Components:

**MonitorServer** (Backend Server)

**Gui Front-end** (C# DotNet application client)

- Monitor Server, C++ OOD, STL, Boost, POSIX Multi-Threads running on  
Linux, (also virtual machine VMware)  
Solaris x86, (also virtual machine VMware)  
Solaris SPARC
- GUI Front-end running on Windows platform
- Communication Layer between Application Server and GUI Client via TCP Sockets:

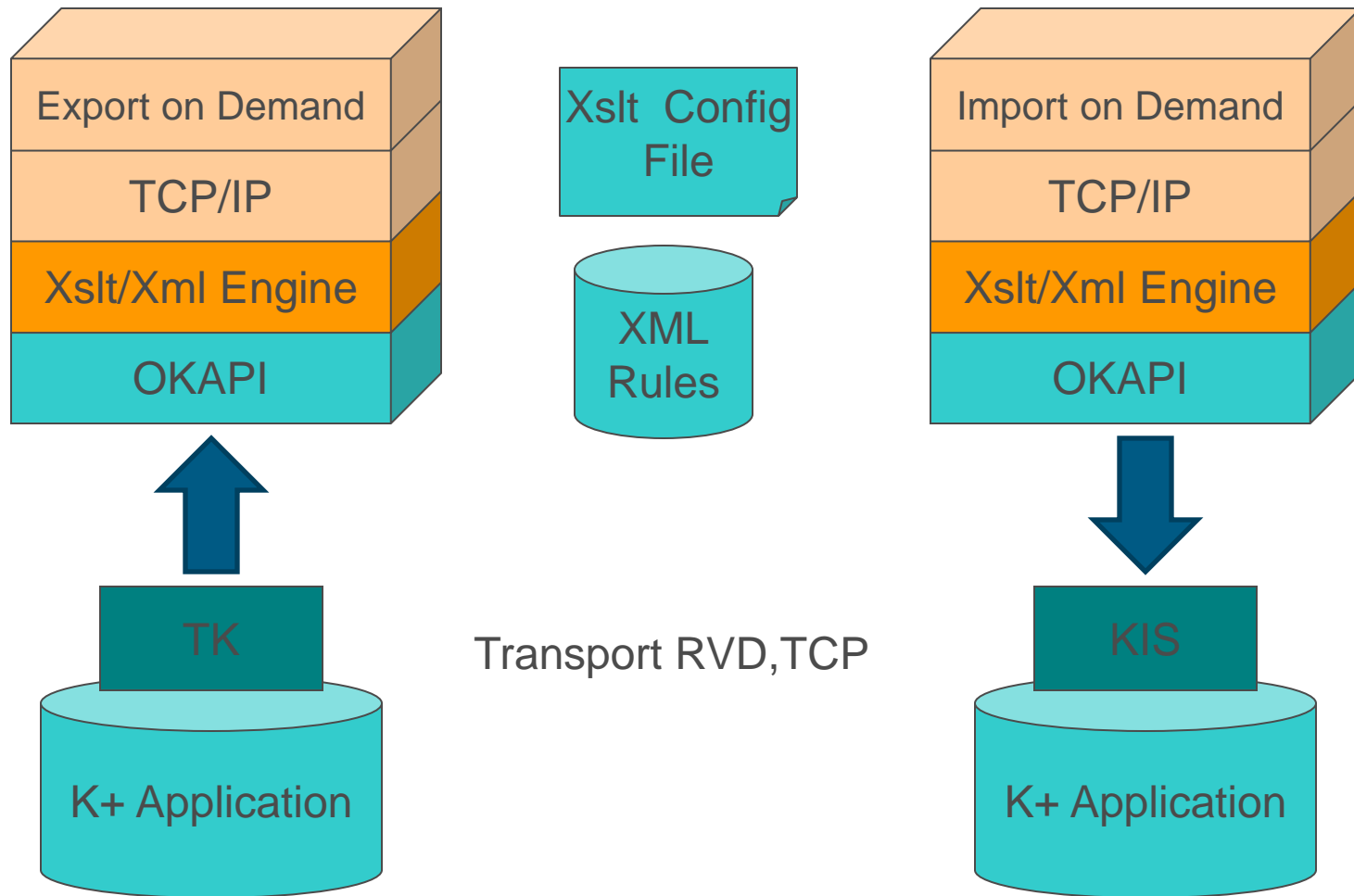
*One socket to process clients requests*

*One socket to listen and receive for the broadcast mvts messages*

# Technologies Market Standard API

- **Kondor+ OKAPI** API to manage the connection with the Tradekast and Kondor Import Server
- **TCP/IP Standard API** to listen on tcp socket and to implement the Reactor Pattern for management the de-multiplexing connection clients (ACE Reactor API, POSA #1)
- **Xerces/Xalan** API at <http://xerces.apache.org/xerces-c/> to support the XML Transformation Engine
- **Log4cpp** Module at <http://log4cpp.sourceforge.net/> for logging and tracing messages by level and severity
- **Boost** Libraries 1.47 at <http://www.boost.org/>
- **FreeTDS** Libraries at <http://freetds.schemamania.org/> to manage the database connection for **Sybase** and **MS Sql Server** Engine

# Monitor Server Workflows



## Meta Data and Workflow

The monitor server is able to manage the following format using internal data structures.

- Kplusfeed format: the Kondor raw messages format exported via the Tradekast
- Internal Xml format: the adapter is able to convert the kplusfeed messages to Xml and from Xml
- Client Xml format: the Xslt Engine will transform the custom xml messages to internal xml format

# Kplusfeed format

{Table;ImportTable;1}  
{TableName;**SpotDeals**;1}  
{Action;I;1}  
{ExportTable;SpotDeals;0}  
{DateFormat;DD/MM/YYYY hh:mm:ss;1}  
{Table;**SpotDeals**;1}  
{DealStatus;V Valid;2}  
{InputMode;I Imported;2}  
{CaptureDate;12/02/2013 14:04:33;5}  
**{TradeDate;29/10/2012;5}**  
{TypeOfEvent;N None;2}  
{Comments;;1}  
{Amount1;555555.000;4}  
{Amount2;-750054.810;4}  
{FixingType;F Free;2}  
{EstimatedAmount;0.000;4}  
{SpotRate;1.350000000;4}  
{SpotMargin;1.000000000;4}  
{ClientSpot;1.350100000;4}  
{CapturedAmount;1 Amount 1;2}  
{IsCLS;N No;2}  
{LastModifDate;12/02/2013 14:04:33;5}  
{Cur1Cur2IsSwapPair;Y Yes;2}  
{Cur1SplitIsSwapPair;N No;2}  
{Cur2SplitIsSwapPair;N No;2}  
{OriginalLocalAmount;-555555.000;4}  
{OriginalLocalRate;1.350000000;4}  
{Cur1Basis;M ACT/360;2}  
{Cur2Basis;M ACT/360;2}  
{SplitCurBasis;M ACT/360;2}  
{LocalCurBasis;M ACT/360;2}  
{CVAPaymentDate;;5}  
{CVAChargeMargin;0.000000;4}  
{CVAChargeAmount;0.000;4}  
{IsCollateralized;N No;2}  
{DownloadKey;#K+Id[813]#,1}

# Internal Xml format

```
<Relation Action="I" ClientDeals_Id="" ClientKey="MONITOR.SpotDeals.813"  
DATA_SOURCE_NAME="MONITOR" ExportTable="SpotDeals" Last ModifDate="" MvtDate=""  
SOURCE_ID="813" TransactionId="813">
```

```
<Table TableName="SpotDeals">
```

```
<Field name="DealStatus" type="enum" value="V Valid"/>
```

```
<Field name="InputMode" type="enum" value="I Imported"/>
```

```
<Field name="CaptureDate" type="date" value="05/12/2012 12:42:31"/>
```

```
<Field name="TradeDate" type="date" value="29/10/2012"/>
```

```
<Field name="TypeOfEvent" type="enum" value="N None"/>
```

```
<Field name="Comments" type="string" value=""/>
```

```
<Field name="Amount1" type="float" value="555555.000"/>
```

```
<Field name="Amount2" type="float" value="-750054.810"/>
```

```
<Field name="FixingType" type="enum" value="F Free"/>
```

```
<Field name="EstimatedAmount" type="float" value="0.000"/>
```

```
<Field name="SpotRate" type="float" value="1.350000000"/>
```

```
<Field name="SpotMargin" type="float" value="1.000000000"/>
```

```
</Table>
```

```
<Table TableName="Users">
```

```
<Field name="Users_ShortName" type="string" value="KPLUS"/>
```

```
<Field name="Users_Name" type="string" value="KONDOR+ ADMINISTRATOR"/>
```

```
</Table>
```

# Transformation Engines

The monitor server implements two Transformation Engines to process the kplusfeed messages.

- The Xslt Transformation Engine to convert Xml client custom format to Internal Xml format managed by the monitor server.
- The Xml Rules Engine to adapt the external messages to kplusfeed format compatible with the Kondor Target System.

The monitor server provides the xsd files to the clients:

- The data.xsd file to describe the Xml Internal Format
- The transform.xsd to describe the rules

# Rules Transformation Engines

The Transformation Engine will apply the rules (xml file based) to the kplusfeed messages before importing to the target Kondor+ system.

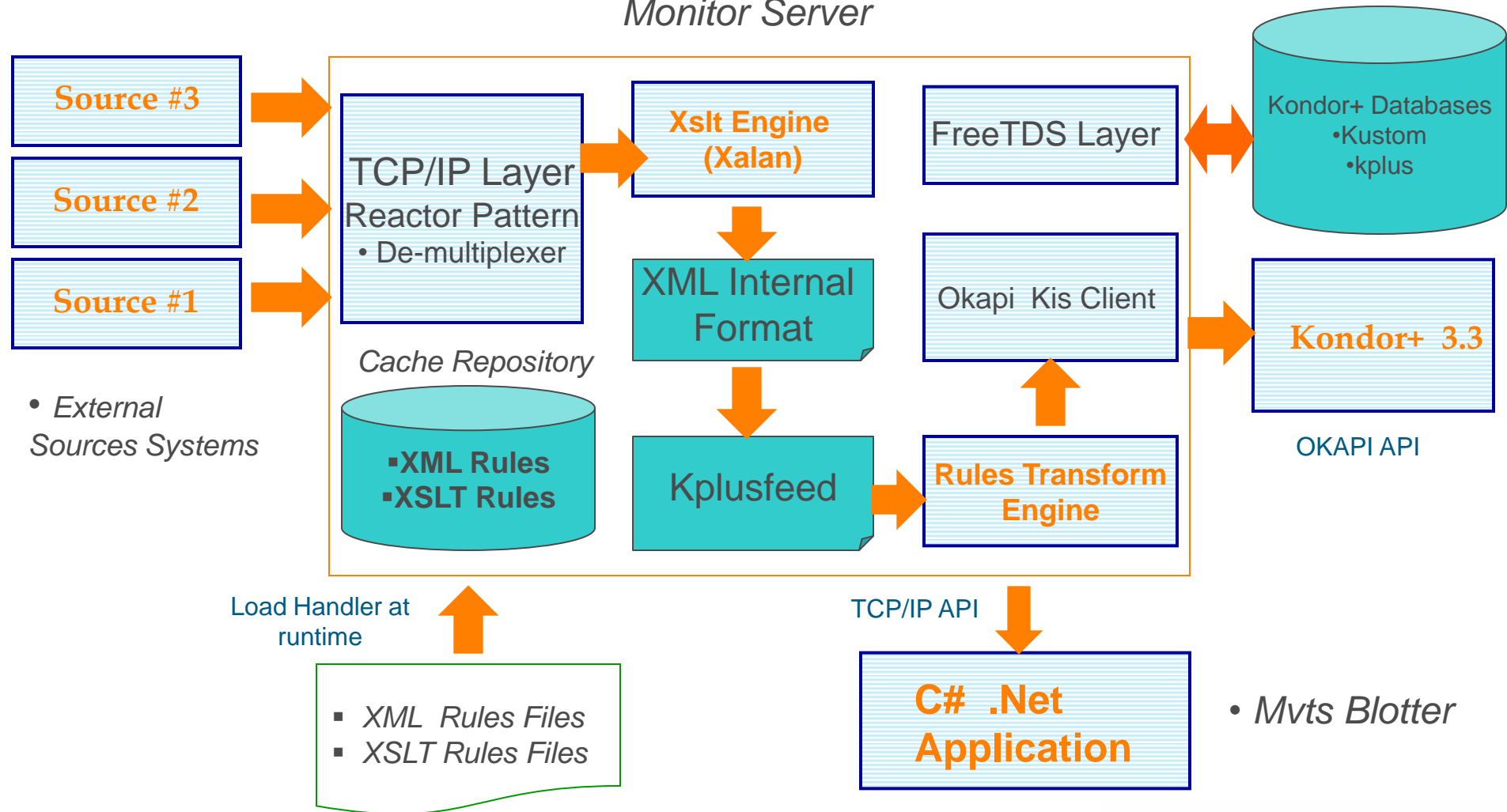
Those xml rules are used for two different purposes:

- to adapt the incoming messages to Kondor+ compatible requirements (missing mandatory fields for Kondor+ 33Sp2)
- to deploy some business logical and conditions to the messages (refer to Folder, Cpty, User other static data).



# Import Handler as Gateway for Integration

## Monitor Server



# Transformation Engines

- The Xslt Engine will load the rules at runtime from file using the LoadXsltRules Handler.

We have set of xslt rules for fixed Source Name.

In this way the Monitor Server could support the xml file import from different sources.

- The Xml Transform Engine will load the xml based rules at runtime using the LoadTransformRules Handler.

We have a set of xml rules for fixed Source Name.

In this way the Monitor Server could support several business logical for different sources.

# IBM MQ Integration

The MonitorServer is fully integrated with IBM MQ to publish the kplusfeed messages into queue in the following mode:

- export to MQ on Demand by Front Users request
- export to MQ in real-time

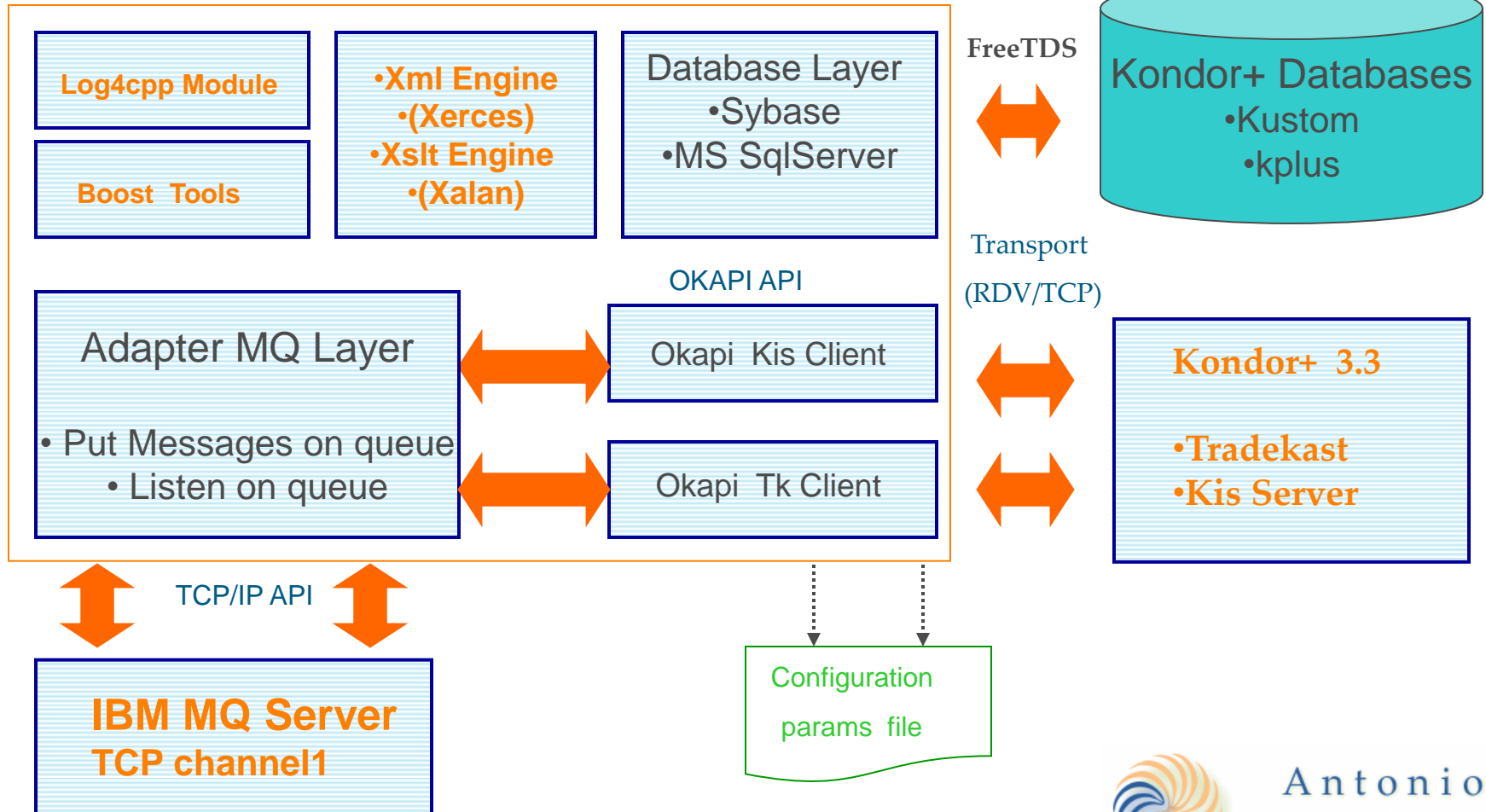
The monitor server is also able to publish in distributed queue at the same time.

The MonitorServer will listen on queue to manage the acknowledge from external system.

It's is possible to customize the listen Callback caller

# Publish Kplusfeed Messages on IBM MQ

Monitor Server (C++)



# Batch Command Tools

List of the command line provided by Monitor Server

- addXsltFile.exe    getDealsInfoTool.exe
- importTool.exe    requestForObjectTool.exe
- updateTool.exe    deleteTool.exe
- getReportMvtsTool.exe    ListenMvts.exe
- requestTool.exe    expToImpTool.exe
- getShortNameTool.exe    loadRulesTool.exe
- testMonitorServer.exe

# RealTime mvts Blotter

## Kondor Monitor Station - [RealTime Mvts Blotter]

File Actions Mvts Monitor ExportOnDemand ImportOnDemand About

### Blotter

MvtDate	Source	TypeOfData	TransId	SourceId	DownloadKey	Action	TypeOfEvent	TypeOfInstr	DealStatus	InputMode	User	Branch	Portfolio	Folder	Cpty	Currency	Pairs	TradeDate	M
11/12/2012 16:57:29	MONITOR	SpotDeals	31097	842	#K+Id[842]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	M
11/12/2012 16:57:28	MONITOR	SpotDeals	31096	841	#K+Id[841]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 16:57:27	MONITOR	SpotDeals	31095	840	#K+Id[840]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 16:57:25	MONITOR	SpotDeals	31094	839	#K+Id[839]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 16:12:12	MONITOR	SpotDeals	31093	835	#K+Id[835]#	D Delete	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 16:12:12	MONITOR	SpotDeals	31092	832	#K+Id[832]#	D Delete	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 15:06:49	MONITOR	SpotDeals	31088	834	#K+Id[834]#	D Delete	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 14:58:54	MONITOR	SpotDeals	31087	838	#K+Id[838]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 14:58:51	MONITOR	SpotDeals	31086	837	#K+Id[837]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 14:58:42	MONITOR	SpotDeals	31085	836	#K+Id[836]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 14:58:40	MONITOR	SpotDeals	31084	835	#K+Id[835]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
11/12/2012 14:58:38	MONITOR	SpotDeals	31083	834	#K+Id[834]#	I Insert	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
▶ 11/12/2012 14:56:21	MONITOR	SpotDeals	31082	833	#K+Id[833]#	D Delete	N None	FXSPOT	V Valid	I Imported	KPLUS	ANTBRANCH	ANTPORT	ANTFOLDER	ENEL		EUR/USD	29/10/2012	
*																			

### Control

Filter

Status

Connected at Broadcast port to Monitor Server

Start

Refresh

Stop

# Comments

- This is just a short draft presentation document to share the Monitor Server project: purposes, technologies and architecture.
- Please feel free to contact me at [adisabat@libero.it](mailto:adisabat@libero.it) for a workshop, demo, further requirements or sponsorship
- Status:
  - Application Server: done
  - GUI DotNet Front-end: Work in progress ...
- (delivery at end of May 2013)

Website: [\*\*www.adisabat.com\*\*](http://www.adisabat.com)



Any questions submit to

[adisabat@libero.it](mailto:adisabat@libero.it)

[www.adisabat.com](http://www.adisabat.com)

**THANK YOU**

