

The background is a vibrant collage of large, colorful letters in shades of blue, green, purple, and pink. Interspersed among these letters are various white icons representing different concepts: a piggy bank, a graduation cap, a fingerprint, a car, a robot, a diamond, a location pin, an eye, a person's face, a speech bubble, a magnifying glass, a pencil, a document, a plane, a person, a group of people, a checkmark, a shopping bag, a compass, a speech bubble with a plus sign, a hand with a plus sign, a plus sign, a Wi-Fi symbol, and a smartphone. The overall theme is digital technology and financial services.

openway

be fast. be smart. be unique.

#1 digital payment software platform provider

OpenWay profile

2017

Prepaid, Debit, Credit, Fuel Card Issuing

Tokenisation / Switch & E-Commerce Gateway

Complex Monitoring: Product view

Openway Service

Konstantin Ryadov / Key SW Engineer / Middleware Team

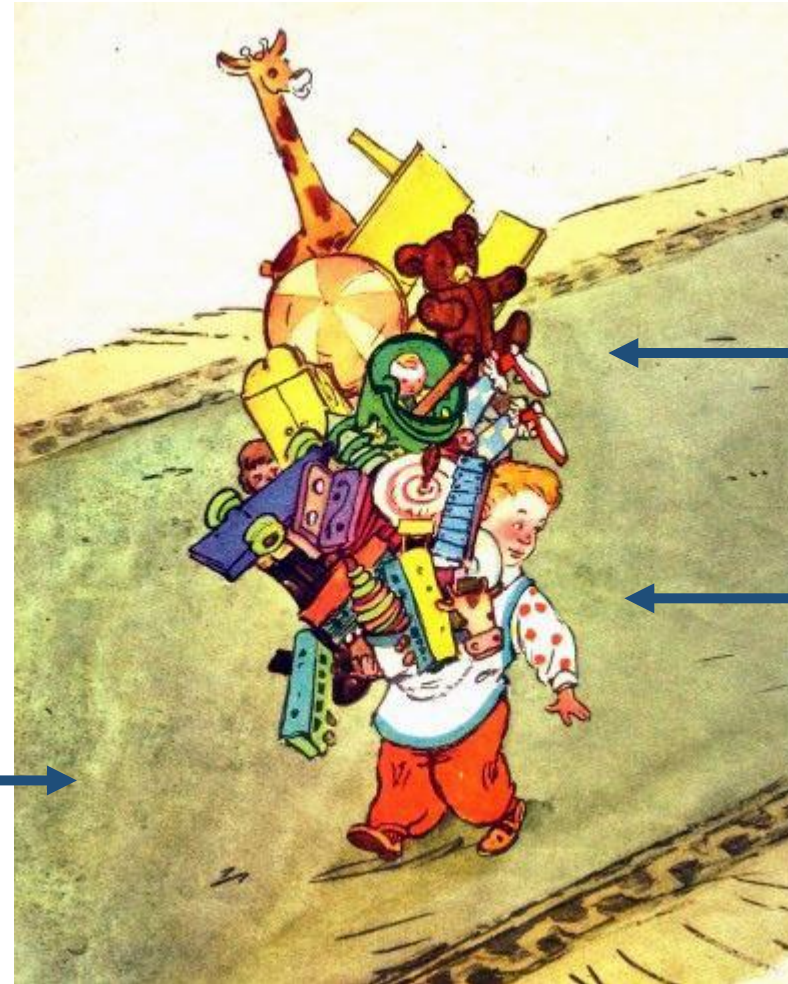
2017

Main Problem – many important* components

Большой карман

Мой знакомый мальчуган
Снят на этой карточке.
У него большой карман
Спереди на фартучке.

С. Я. Маршак



← Components

← Client

Roadmap →

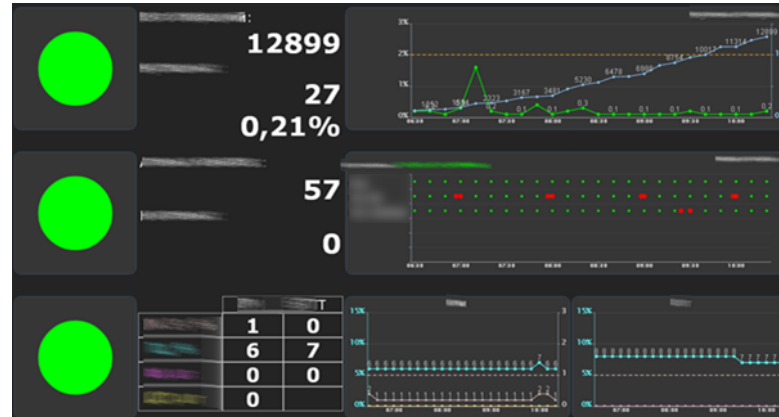
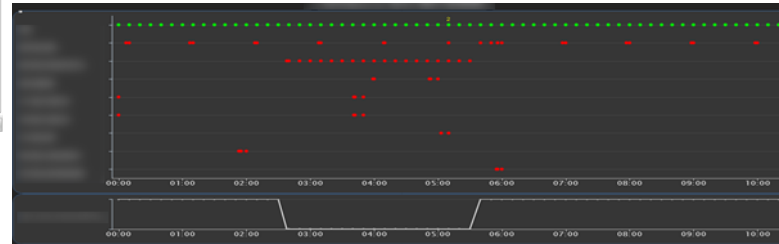
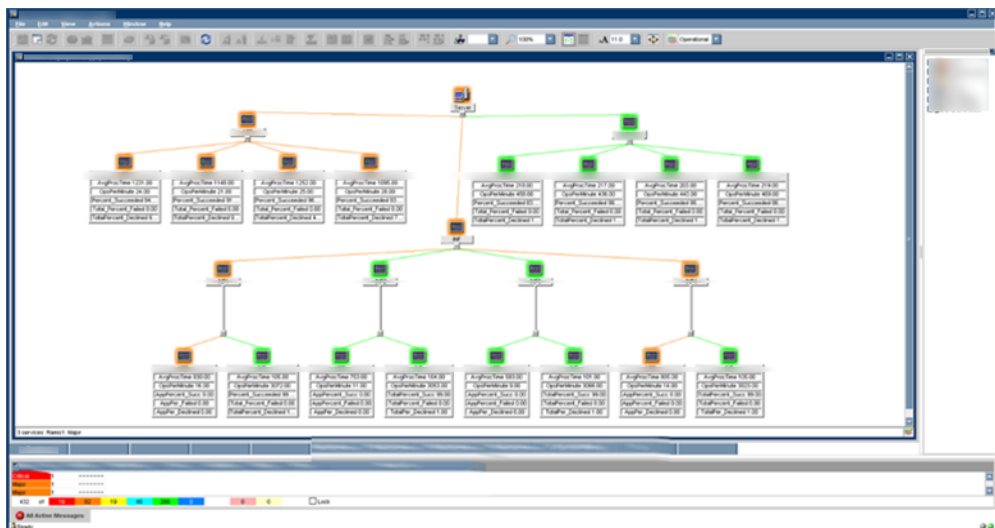
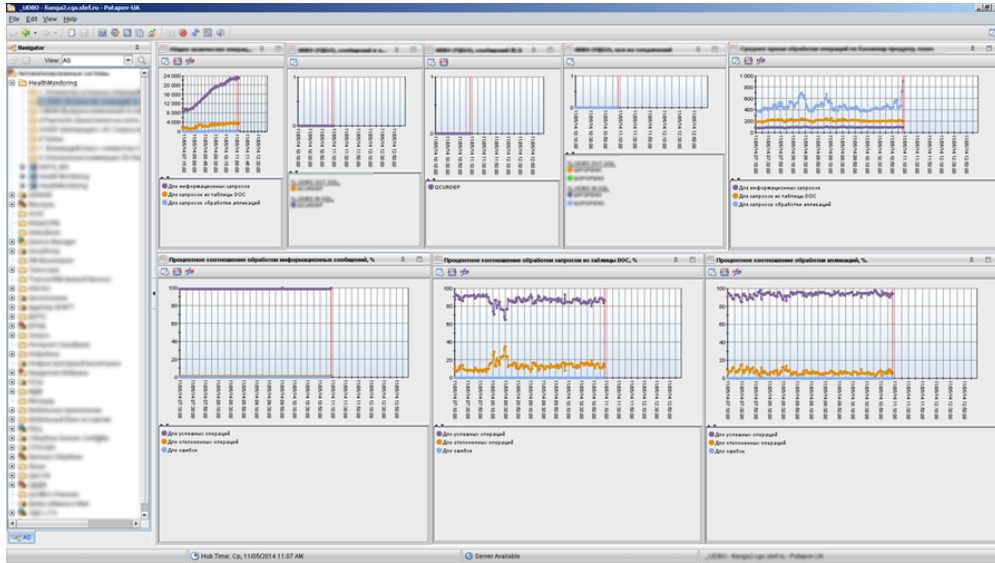
* Important component must be under monitoring

Other issues in Openway Service context

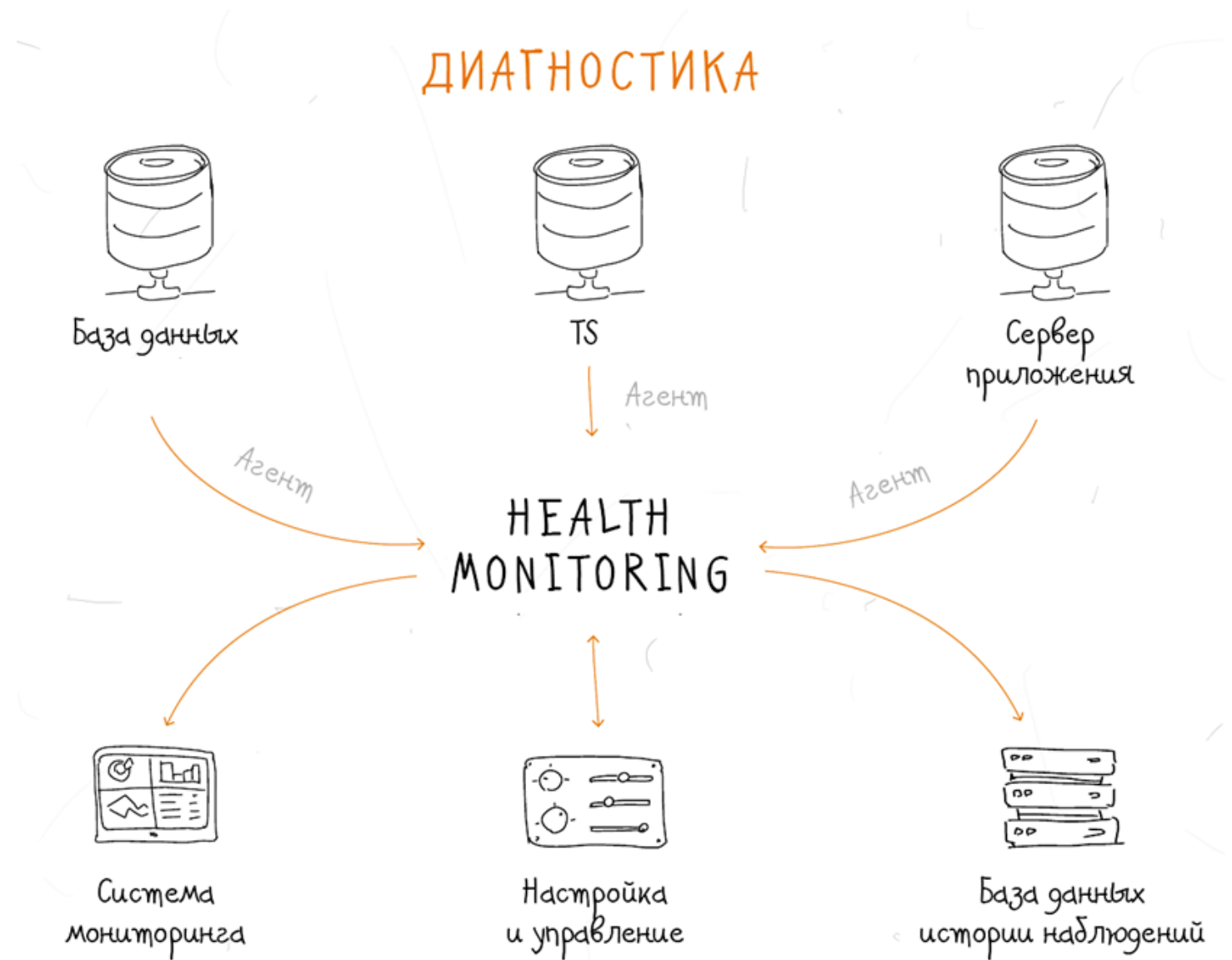
- 100+ independent/different banking-oriented clients
- **AIX Power** / **Solaris SPARC** / **Linux/Windows x86_64***
- Isolated (no internet) / mixed environment
- **No client resources** to “monitoring deep dive”
- Linked external monitoring resources (DB / MQ / WS / ...)
- **No client resources** to setup/maintain it.

* - mixed environment

Monitoring Wanted



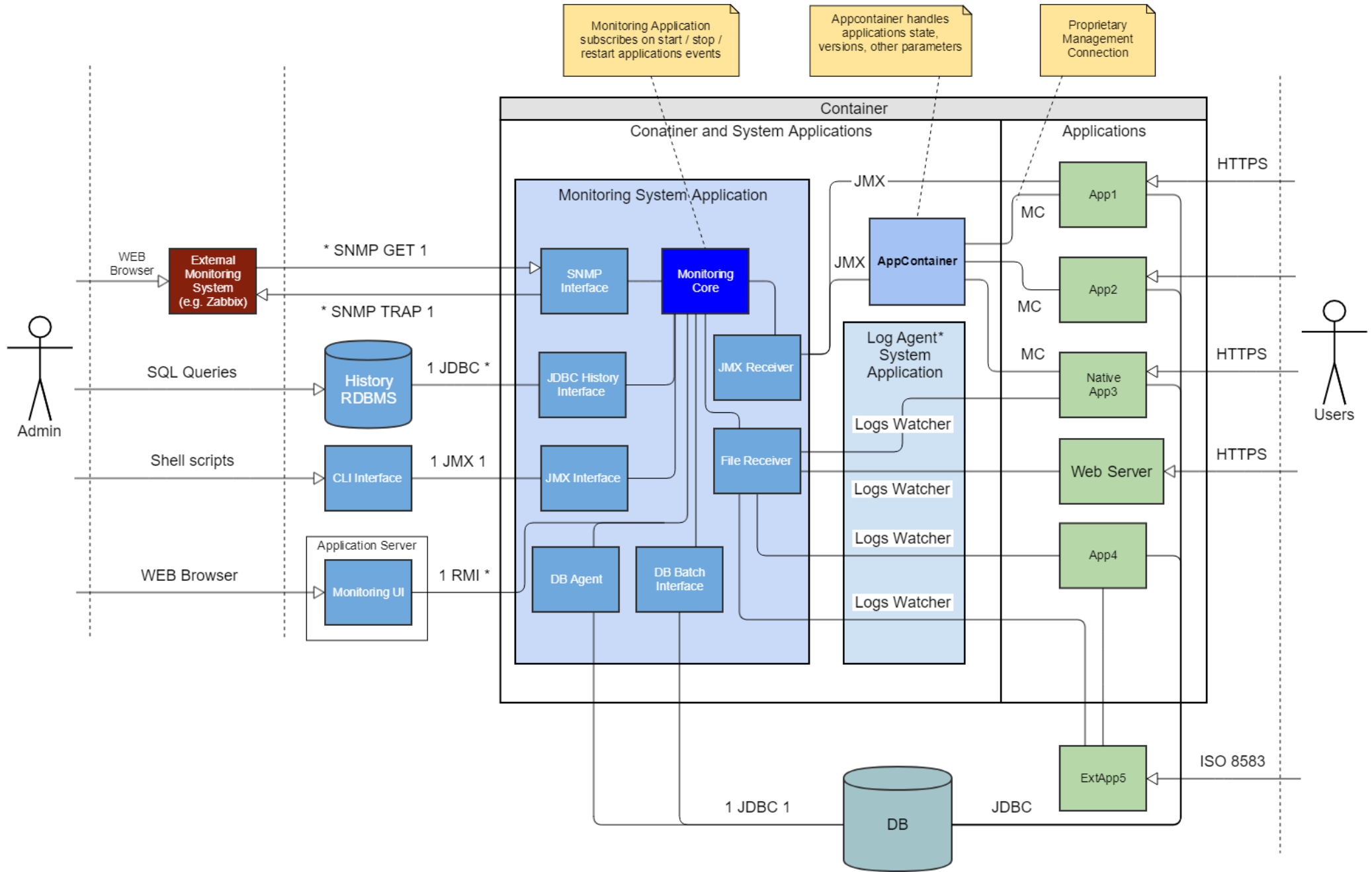
Handsome Architecture



How?



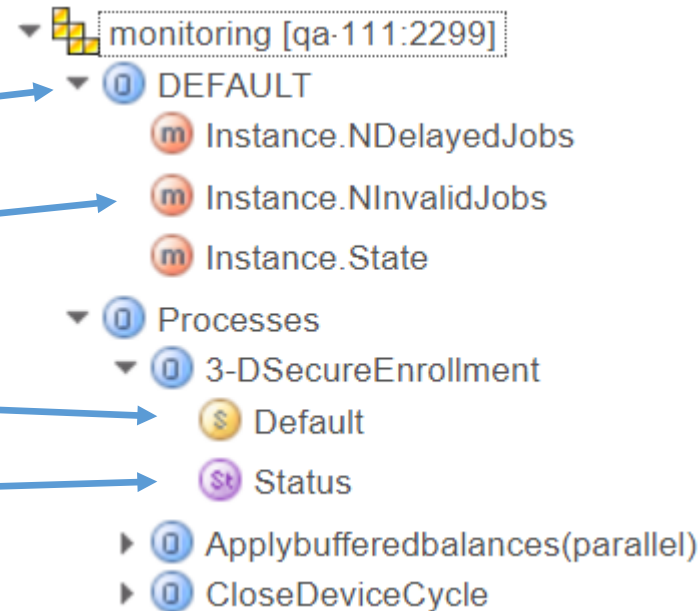
Real Architecture



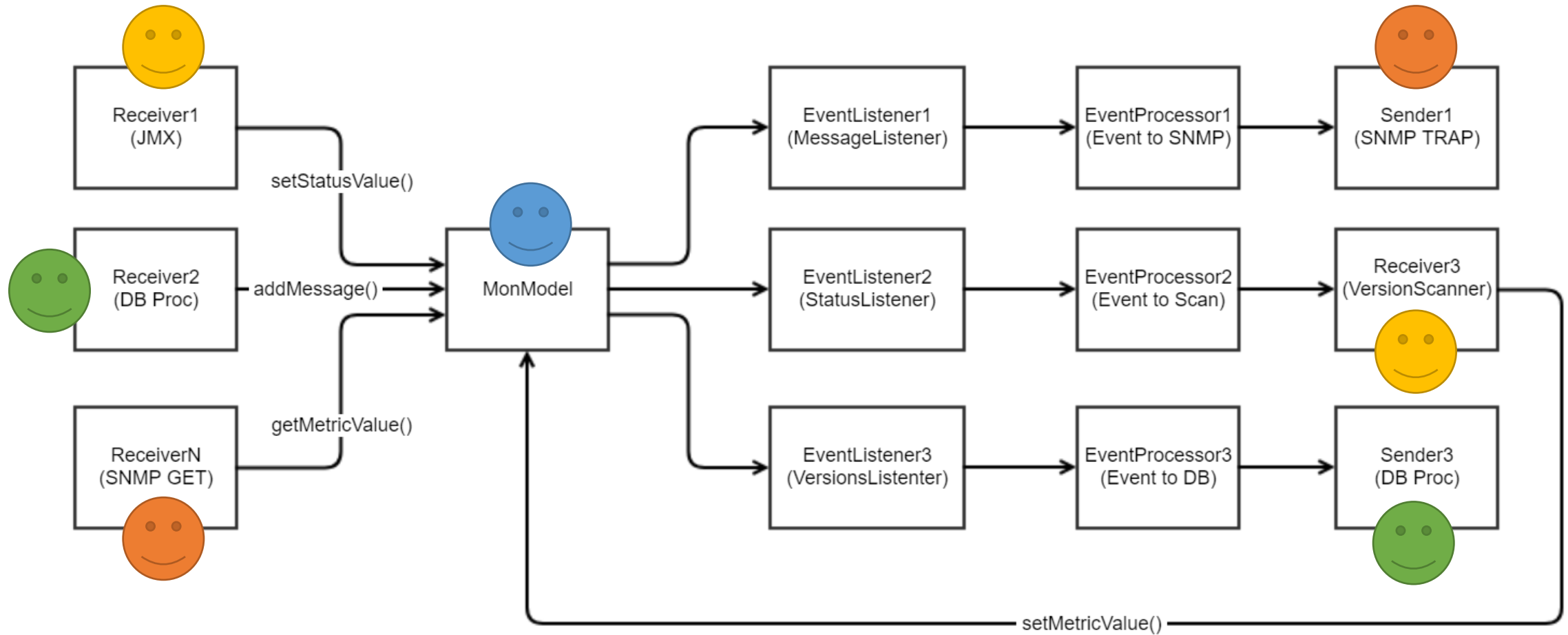
Monitoring domain – monitoring objects

Monitoring:

- Object
- Metric
- Stream (of messages)
- Status



Core – domain, API, SPI and implementation



Spring 4 on Groovy 2.4 in 2015.2

Start with:

- Tech holy war architect question – what technology/platform?
- Configuration? Distributed configuration? Zero configuration?
- Update in client case with no DevOps / ProdOps?

End with:

- Change log-level for debug reasons without restart monitoring?

DISCUSS all non-functional requirements with your team!



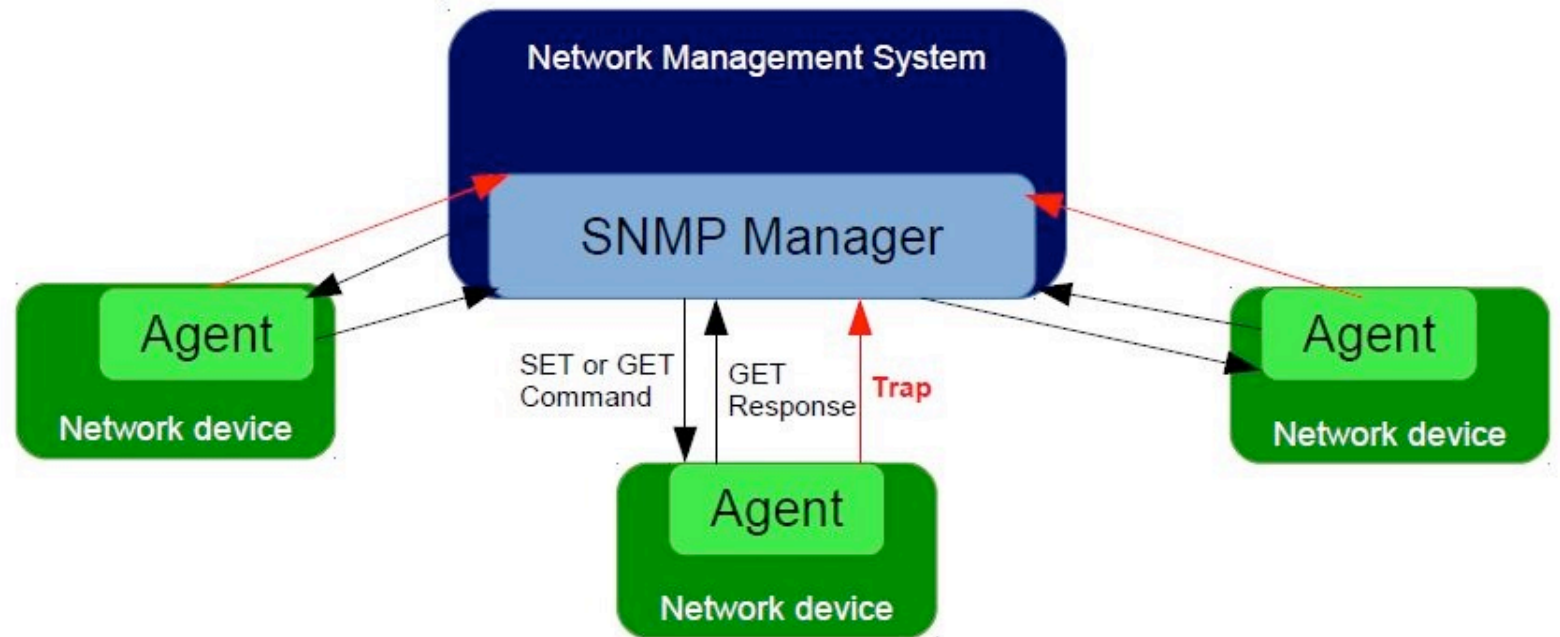
SNMP / Static MIB

SNMP: Simple Network Management Protocol

MIB: Management Information Base

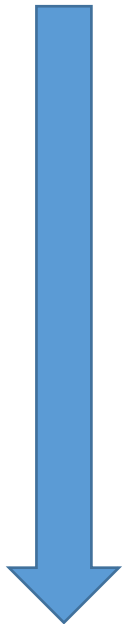
Examples:

1. [HP-LASERJET](#)
2. [ILOM - E24707 01](#)
3. [CentOS 7](#)



Dynamic MIB

1. Add mon objects
2. Exec monitoring mib.generate
3. OPENWAY-SNMP-MONITOR.mib:



The screenshot shows a monitoring interface with a 'Monitoring tree' on the left and a 'Properties group' on the right. The tree view shows a hierarchy of monitoring objects, with 'OpsPerSecond.Declined' highlighted in red. The properties panel on the right shows the following details for the selected metric:

- Metric**
 - SNMP OID: 1.3.6.1.3.566.2.2.58.25.4.1.2.1.4
 - Path: /01/E_COMMERCE/Input/AllMsg/OpsPerSecond.Declined
 - Reg date: 2017-07-11T21:30:53.218+03:00
 - Enabled: true
- Metric value**
 - Type name: OpStat:OpsPerSecond
 - Measure unit: Per Second [s]
 - Value: 0.0
 - Valid from: 2017-07-11T21:25:00.000+03:00
 - Valid to: 2017-07-11T21:30:00.000+03:00
 - Expire period: 10 minutes
- Show raw value:
- Buttons: Refresh value, Delete

```
--1.3.6.1.3.566.2.2.58.25.4.1.2.1.4 server/01/E_COMMERCE/Input/AllMsg/OpsPerSecond.Declined  
ns-01-e-commerce-input-allmsg-opspersecond-declined OBJECT-TYPE  
SYNTAX INTEGER -- multiplier: 1000
```


SNMP Walk on dynamic monitoring tree

```
/usr/bin/snmpwalk -v 1 -c public localhost:1161 1.3.6.1.3.566.2.2.52
```

...

```
SNMPv2-SMI::experimental.566.2.2.52.1.1.0 = STRING: "Database connection pool opening..."
```

```
SNMPv2-SMI::experimental.566.2.2.52.2.1.1.0 = ""
```

```
SNMPv2-SMI::experimental.566.2.2.52.2.1.2.0 = STRING: "Process Access Server : Active. Parameters: HOSTNAME=server1;CONNECTION_
```

```
GROUP_ID=main;CONNECTION_POOL_ID=WAY4U Kernel Notifications;"
```

```
SNMPv2-SMI::experimental.566.2.2.52.2.2.1.0 = ""
```

```
SNMPv2-SMI::experimental.566.2.2.52.2.2.2.0 = STRING: "Application WAY4Processes/ApplyProductChanges is C"
```

SNMP4J

- <http://www.snmp4j.org/>
- Really multiplatform
- SNMPv1/v2c/v3
- UDP/TCP
- Security in SNMPv3

- SNMP
 - GET/TRAP
 - WALK



CLI and UI

Monitoring JMX CLI returned:

```
{  
  "monObject": "MonStatus",  
  "path": "ATM1/Status",  
  "numId": 1,  
  "regDate": 1469097915942,  
  "updDate": 1469097915942,  
  "monObjectType": "appserver-app",  
  "enabled": true,  
  "statusValueType": "ApplicationStatusValue",  
  "statusValue": {  
    "finishTime": 1499627511786,  
    "statusValue": "STOPPED",  
  },  
  "startTime": 1499627511786,  
  "statusValueClass": "ApplicationStatus"  
}
```

The screenshot shows a web-based monitoring interface. At the top right, it says "Logged as hmadmin" with a "Logout" link. Below this are two tabs: "Monitoring tree" and "Properties group".

The "Monitoring tree" tab is active, displaying a hierarchical tree of monitoring objects. The root is "Unknown [01:1099]". Underneath is "monitoring [-9:2299]". Under "monitoring" is "ATM1", which has a "Default" icon and a "Status" icon. The "Status" icon is highlighted. Below "ATM1" are several other monitoring objects: ATM2, ATM3, ATM4, ATM5, DEFAULT, Mojo, Sch, W4W_internal, appserver, authgate, ccc, container, database, internal, logagent, m2_web_console, monitoring, monitoring_ui, monitoring_ui1, and smsh.

The "Properties group" tab is also visible, showing the "Status" of the selected "ATM1/Status" object. It displays the following information:

- Status**
- SNMP OID:** 1.3.6.1.3.566.2.2.50.9.1
- Path:** ATM1/Status
- Reg date:** 2016-07-21T13:45:15.942+03:00
- Enabled:** true

Below this, there is a "Status value" section:

- Status value**
- Value:** stopped
- Reg date:** 2017-07-09T22:11:51.786+03:00
- Expire period:** Unexpired

At the bottom of the "Status value" section, there is a "Show raw value" checkbox (which is unchecked) and two buttons: "Refresh value" and "Delete".

JMX Beans as CLI and licensing

```
@ManagedOperation(description = 'Register process statuses for last ${period} day(s)')
@ManagedOperationUtilityDescription("cards.register.statuses")

void registerStatuses(String period) {
    if (!dbAgentsExecutor.componentStarted) {
        ...
        log.debug("Registered process statuses for last $period day(s)")
    }
}
```

```
[way4@srv bin]$ ./monitoring
```

```
Monitoring JMX CLI
```

```
Usage: monitoring [<group_name>.]<command> [arguments]
```

```
Please use --help as argument to see the full syntax of the command
```

```
Available commands:
```

```
Group 'historyDatabase' - History database management
```

```
    update.history.database      -    Update history database
```

```
Group 'certificate' - Certificate management
```

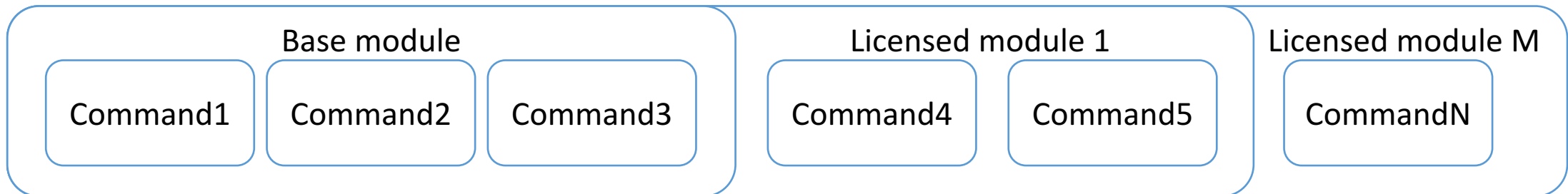
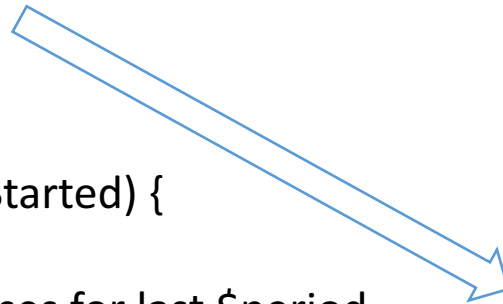
```
    certificate                  -    Manage certificates
```

```
Group 'remoteSqlExecutionService' - SQL Statement Executor on WAY4 CARDS schema
```

```
    cards.register.statuses - Register process statuses for last ${period} day(s)
```

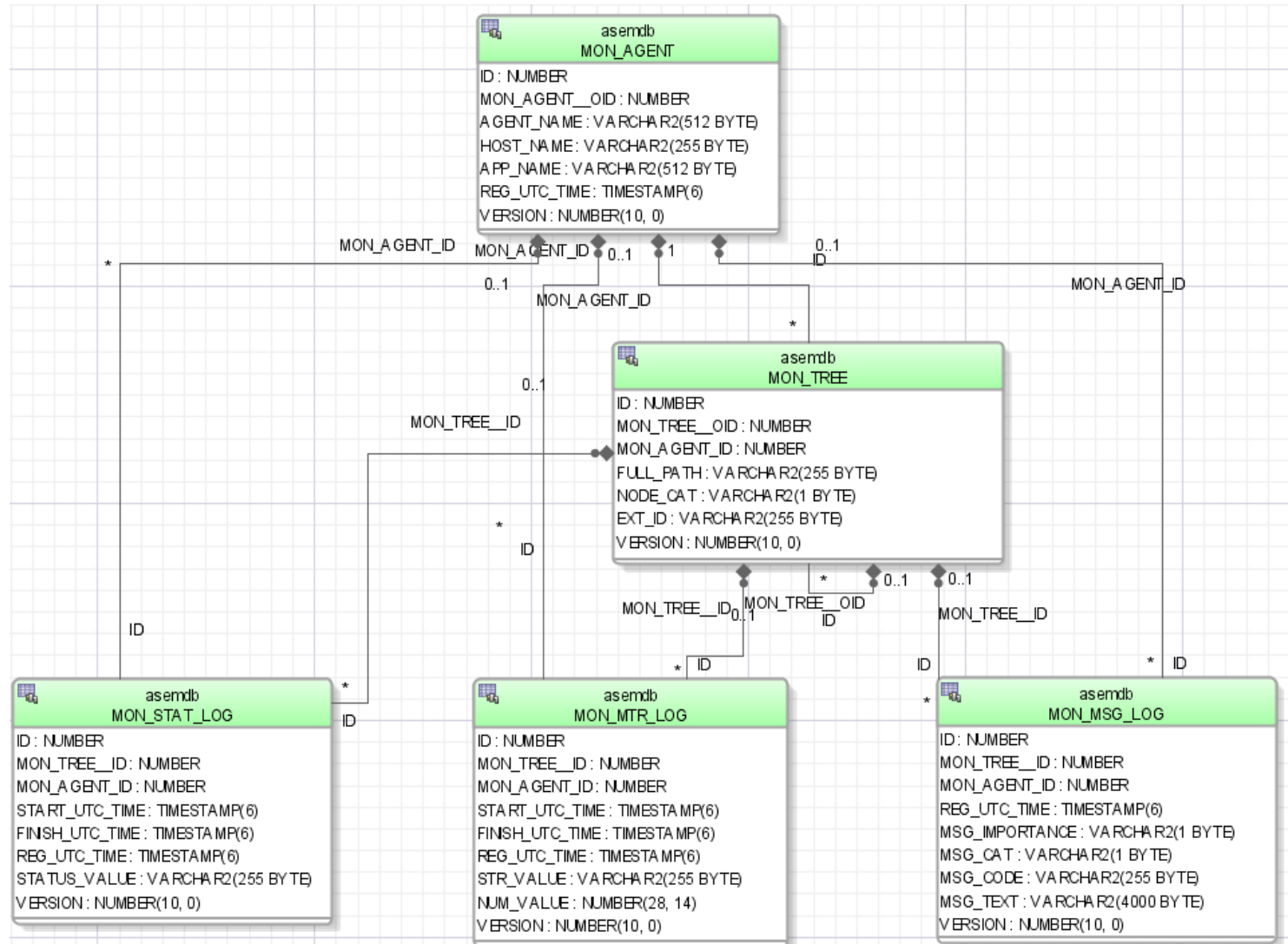
```
Group 'jmxReceiver' - Application server JMX gate versions
```

```
...
```



Mini-History DB on Liquibase + Hibernate

- 5 tables
- Hibernate
- Liquibase
- Typical queries
- Row / partition HSK



```
select mon_msg_log.reg_utc_time,
       msg_importance, msg_cat,
       msg_code, msg_text,
       agent_name
from mon_msg_log, mon_agent
where
/* ERROR("E"), WARN("W"), ... */
   msg_cat in ('E', 'W') and
  mon_tree__id in ( ...
```


Deployment

- “Bundled with” main component (deploy with main component)
- Standalone distro is also available

```
Left File Command Options Right
<- ~/autoDeployDistr .[^> <- ...zip/uzip://appserver/temp/sysapps .[^>
.n Name Size Modify time .n Name Size Modify time
/.. UP--DIR Jul 13 02:34 /.. UP--DIR Jul 12 09:08
*WAY4Applicat-parc-Install 1874915 Jul 12 09:04 logagent-2.0.860.zip 19359K Jul 12 09:08
appserver-an-1.6.2122.zip 2820680 Jul 12 11:37 m_agent-1.0.1923-app.zip 28981K Jul 12 09:08
appserver-ap-218-dist.zip 26712K Jul 12 11:37 m_tools-1.0.1923-app.zip 10091K Jul 12 09:08
appserver-ap-218-dist.zip 14096K Jul 12 11:37 m_web_consol~1.0.1923.war 20934K Jul 12 09:08
appserver-ap~-1.7.218.zip 524237 Jul 12 11:37 monitoring-1.0.1148.zip 73151K Jul 12 09:08
appserver-ce-1.6.2122.zip 21208 Jul 12 11:37 monitoring_u-1.0.1148.war 60065K Jul 12 09:08
appserver-co-1.0.1923.zip 1678 Jul 12 11:37 web_console-1.6.2122.war 25533K Jul 12 09:08
appserver-do-1.6.2122.zip 5040775 Jul 12 11:37
appserver-in-is-sparc.zip 1204877 Jul 12 11:37
appserver-jd~-30-dist.zip 186116K Jul 12 11:37
appserver-le-1.6.2122.zip 22405 Jul 12 11:37
appserver-m ~1.0.1923.zip 3856330 Jul 12 11:37
appserver-st-1.6.2122.zip 560 Jul 12 11:37
appserver-st-1.6.2122.zip 1742 Jul 12 11:37
appserver-sy-1.6.2122.zip 238152K Jul 12 11:37
appserver-sy-1.6.2122.zip 101805 Jul 12 11:37
243,867,985 bytes in 1 file 156,237,019 bytes in 3 files
appserver-sysapps-m_agent-1.6.2122.zip monitoring-1.0.1148.zip
276/346 (79%)
```

Hint: If your terminal lacks functions keys, use the ESC+number sequence.

```
~/autoDeployDistr$
```

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit



Product Quality

1. CD



2. Code coverage

Overall Coverage Summary

Package	Class, %	Method, %	Line, %
all classes	87.7% (512/584)	84.1% (1215/1444)	83% (7889/9503)

3. Tests

▼ MAgent AIX 7 gradle tests | ▼
#1.6.2122 Tests passed: 248 | ▼

4. Deploy to infrastructure / production

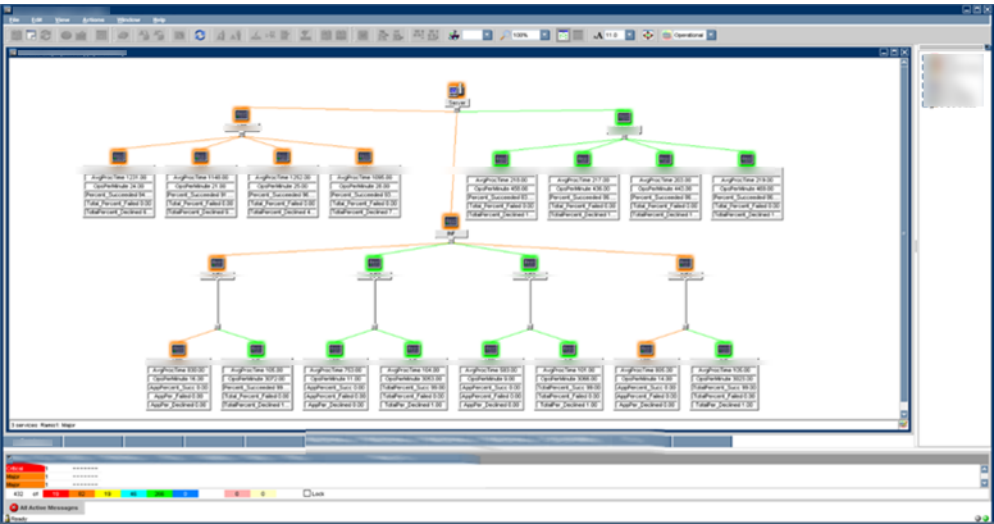
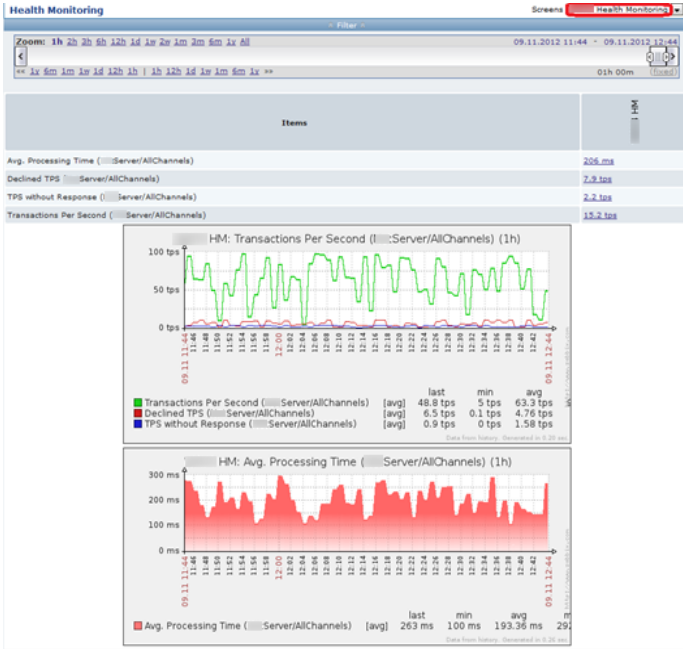
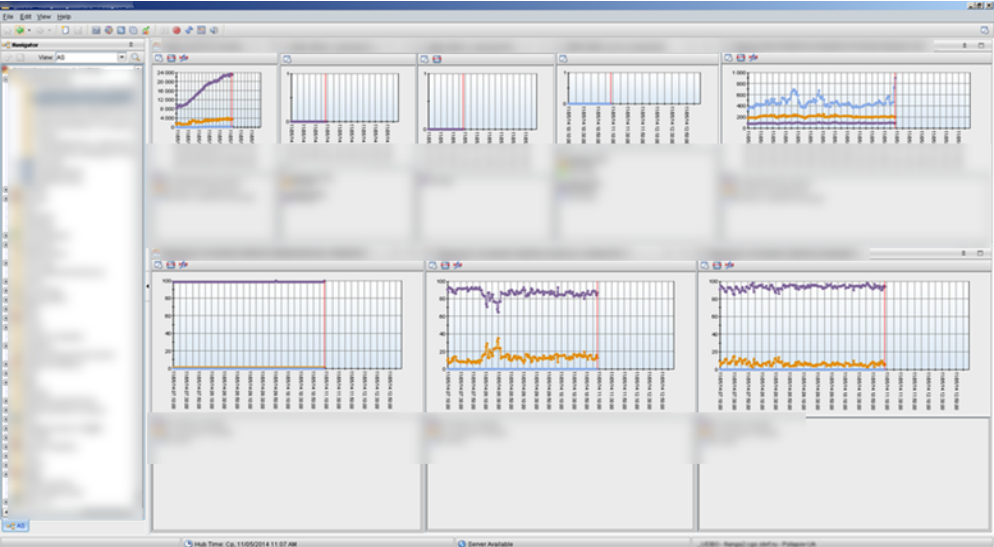
LET'S DEPLOY ALL PRODUCTS



What would be better

- Use Kotlin for development – it is stable now
- Look at Prometheus - <https://prometheus.io/>
- Look at Elasticsearch / Splunk for logs/events search
- Get rid of RMI in heterogeneous environments
- Get rid of heterogeneous environments 😊
- Look at SCM (Ansible, Chef, Puppet, SaltStack) for large deployments

Result



A screenshot of a monitoring tree and properties group interface. The 'Monitoring tree' shows a hierarchy of nodes, with several nodes highlighted in red and marked with warning icons. The 'Properties group' shows details for a selected stream (message):

Stream (message)

- SNMP OID: 1.3.6.1.3.566.2.2.85.17.1.1.1.2.1
- Path: Node/.../Inc
- Reg date: 2016-07-26T16:48:44.086+03:00
- Enabled: true

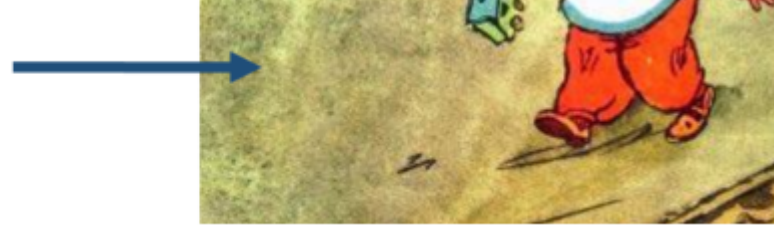
Stream value

- Code: WAY4R_03
- Message: W4R Apply latency is greater than 10
- Importance: MAJOR
- Category: WARNING
- Reg date: 2016-11-07T20:45:34.465+03:00

Buttons: Refresh value, Delete

Summary

- Remember your Roadmap
- Define domain
- Do/implement it in parallel
- Do simple
- Complex test it / think about continuous delivery



100+ independent/different banking-oriented clients

We've done it!





thank you!
questions

www.openwaygroup.com

www.facebook.com/Openwaygroup

www.twitter.com/openwaygroup