ZABBIX 2014 Conference

freenet digital

Automated and Scalable Configuration Management for Zabbix in a Large Java EE Environment

Who am I?

- more than 10 years experience in various areas of Enterprise Java
- 6 years of work for different consulting companies
- JBoss support and training pioneer
- strategy and architecture team @ freenet digital
- technical guidelines, software infrastructure
- Application Monitoring is one part of our work
- settled near Berlin with my family (2 kids)
- passionate marathon runner

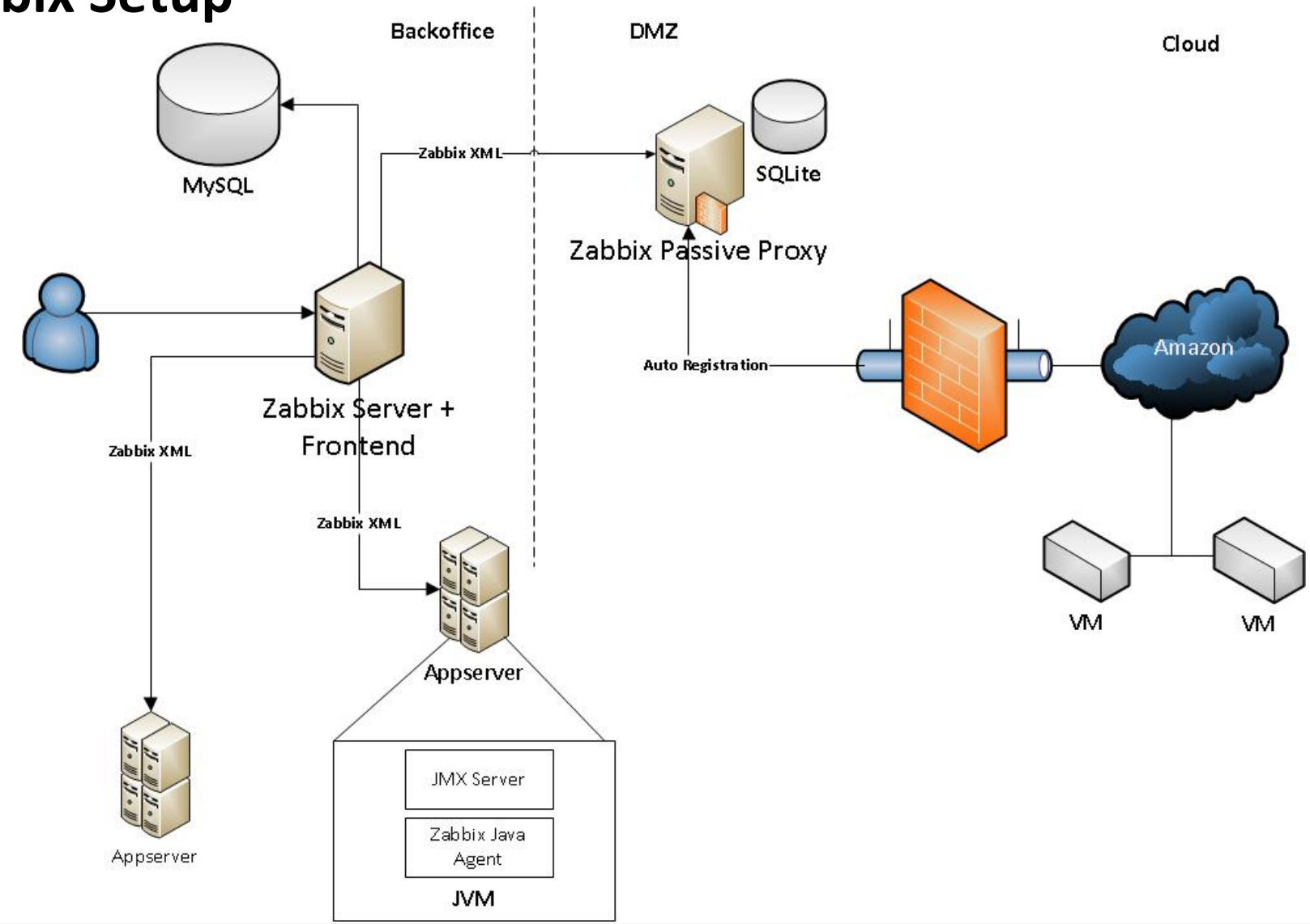
Company Facts And Figures

- freenet digital is part of freenet Group, the largest network-independent telecommunications provider in Germany
- leading global provider of next generation entertainment content and services for the digital consumer
- 300 professionals in offices in Berlin (Germany) and Los Angeles (USA)
- freenet digital's brands include mobile entertainment brands Jamba and Jamster, social dating community iLove and the mobile ad network solution Motility Ads

Agenda

- current monitoring configuration process and its disadvantages
- requirements for a process automation and scalability
- Zabbix API as a proper solution
- implementation within a custom management tool
- how we tackled performance issues in a steady growing platform
- experiences and outcome of the project
- optional: experiences with migration from Zabbix 1.8 to 2.2

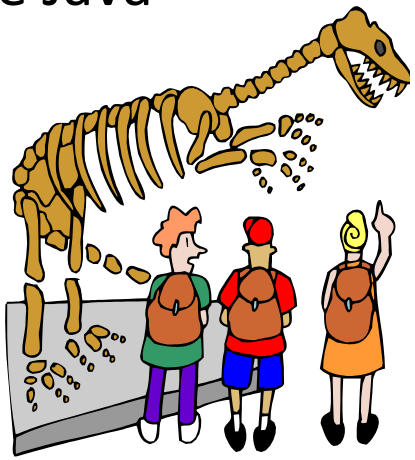
Zabbix Setup



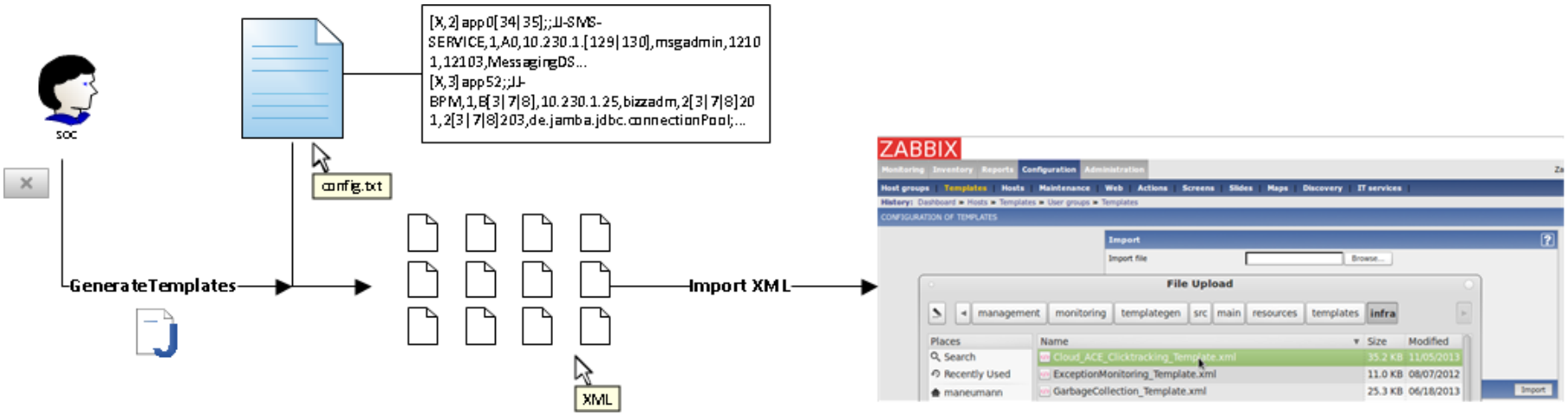
freenet digital

Zabbix History @ freenet

- introduced in 2008
- update path: 1.6.2 → 1.8.2 → 1.8.5 → 2.2.5 (July 2014)
- staged installations (laboratory, integration, live)
- JVM monitoring has the highest priority
  - platform is almost 100% based on Enterprise Java
- sponsored development



Monitoring Configuration Status Quo



- manual process steps
- error-prone
- high maintenance efforts
- template import is very slow
  - takes more than 5min for big templates

freenet digital

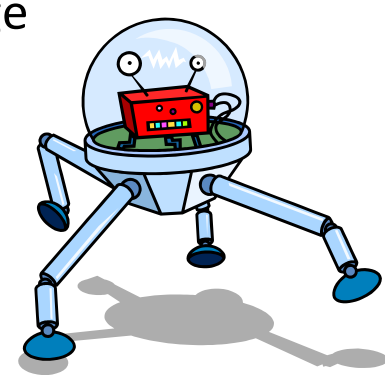
Monitoring Configuration

```
<?xml version="1.0"?>
<zabbix_export version="1.0" date="25.09.08" time="09.32">
    <hosts>
        <host name="JBOSS_${TEMPLATE}_TEMPLATE">
            <useip>0</useip>
            <dns></dns>
            <ip>0.0.0.0</ip>
            <port>10050</port>
            <status>3</status>
            <groups>
                <group>${TEMPLATE_GROUP_NAME}</group>
            </groups>
            <host_profiles_ext>
            </host_profiles_ext>
            <items>
                <item type="0" key="jmx[jboss.system:type=Server][VersionNumber]" value_type="1">
                <#list EXCEPTION_TYPES as EXCEPTION_TYPE>
                    <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=ExceptionMonitor][${EXCEPTION_TYPE}]" value_type="3">
                </#list>
                <#list USER_EXCEPTION_TYPES as USER_EXCEPTION_TYPE>
                    <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=ExceptionMonitor][${USER_EXCEPTION_TYPE}]" value_type="3">
                </#list>
                <#list JAMBA_EXCEPTION_TYPES as JAMBA_EXCEPTION_TYPE>
                    <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=ExceptionMonitor][${JAMBA_EXCEPTION_TYPE}]" value_type="3">
                </#list>

                <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=ExceptionMonitor][javax.transaction.TransactionRolledbackExceptio
                <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=ExceptionMonitor][java.security.InvalidKeyException]" value_type=
                <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=ExceptionMonitor][ExceptionCount]" value_type="3">
                <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=ExceptionMonitor][ExceptionMap]" value_type="1">
                <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=LogMonitor][LogErrorCount]" value_type="3">
                <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=LogMonitor][LogFatalCount]" value_type="3">
                <item type="0" key="jmx[com.jamster.infra.appserver.monitoring:service=LogMonitor][LogWarnCount]" value_type="3">
                <#list DB_JNDI_NAMES as DB_JNDI_NAME>
                    <item type="0" key="jmx[jboss.jca:service=ManagedConnectionPool,name=${DB_JNDI_NAME}][InUseConnectionCount]" value_type="3">
```

Automation Requirements

- Overall goal: all administrative tasks can be done without the Zabbix frontend (read-only access)
- reduce the maintenance efforts by ~70%
- templates can be created and assigned to different abstraction levels
  - use defaults as much as possible
- all changes are recorded and can be rolled back
- change only single parameters (e.g. change the threshold of a trigger)
- operations team should not require deep Zabbix knowledge
- easy configuration process
- better performance

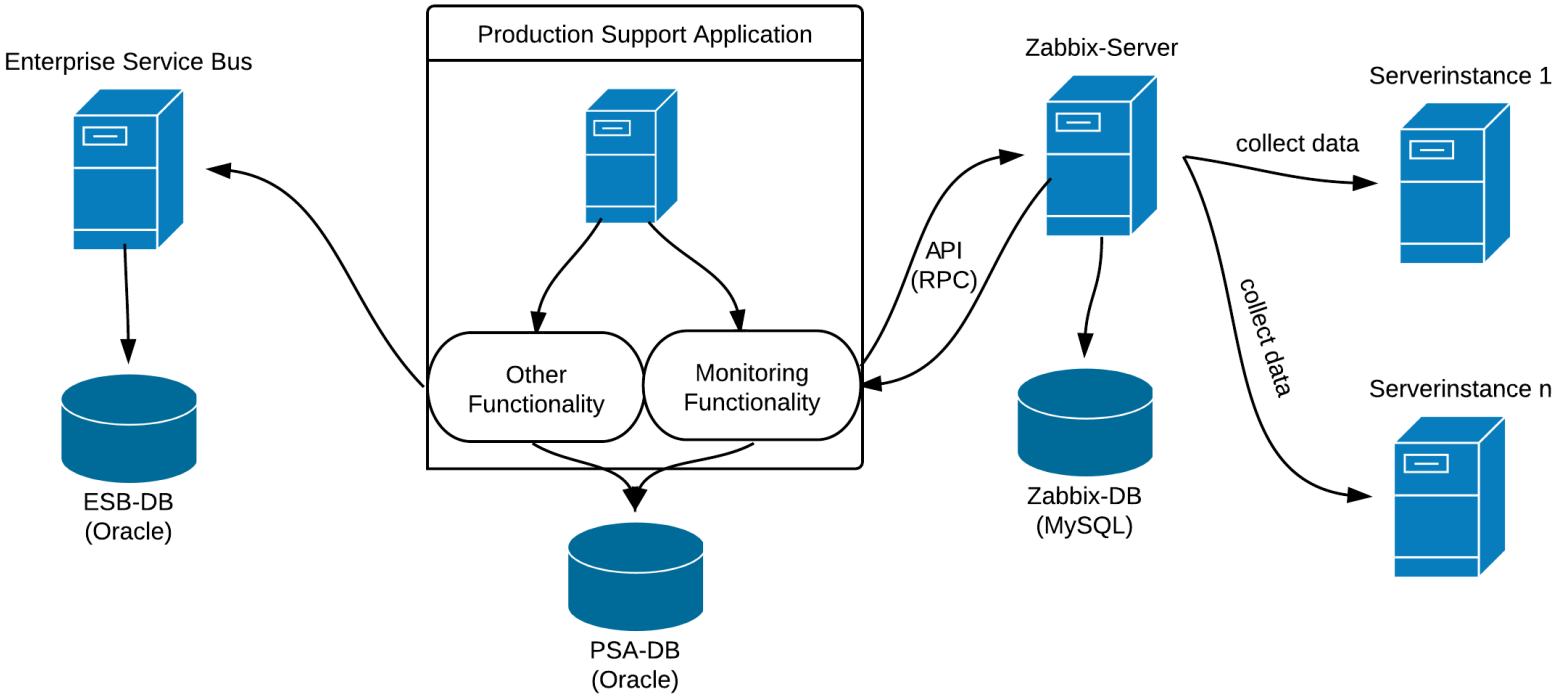


Automation Requirements

- custom management application is existing (PSA)
  - server management
  - service repository
- decision: integration of the Zabbix configuration process
- CRUD on items/triggers/macros
- synchronization of hosts created in PSA with current Zabbix configuration
  - monitoring state
  - activate/deactivate monitoring
- no need to edit text files, xml files, templates or other error-prone configuration locations

Zabbix API

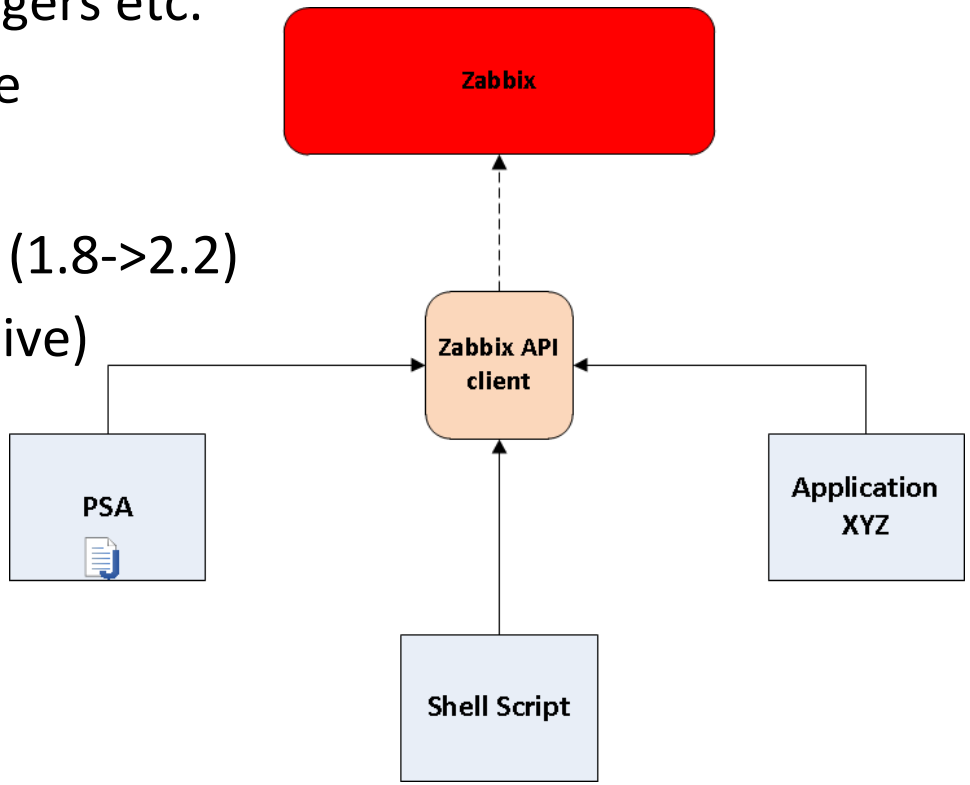
- API was introduced with Zabbix 1.8
- **enterprise feature** required for integration purposes
- work on automation project started 2011 with Zabbix 1.8



freenet digital

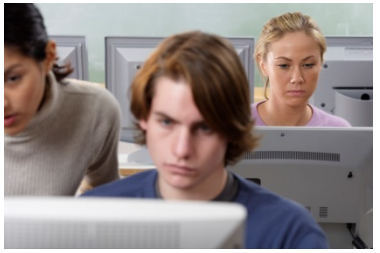
Implementation

- started with a basic implementation of a Java api client
- client can be reused in different applications
  - modules for hosts, items, triggers etc.
- no hardcoded access to api inside the Zabbix clients!
- consistent versioning of changes (1.8->2.2)
- configurable target server (test, live)
- generic exception handling for all clients



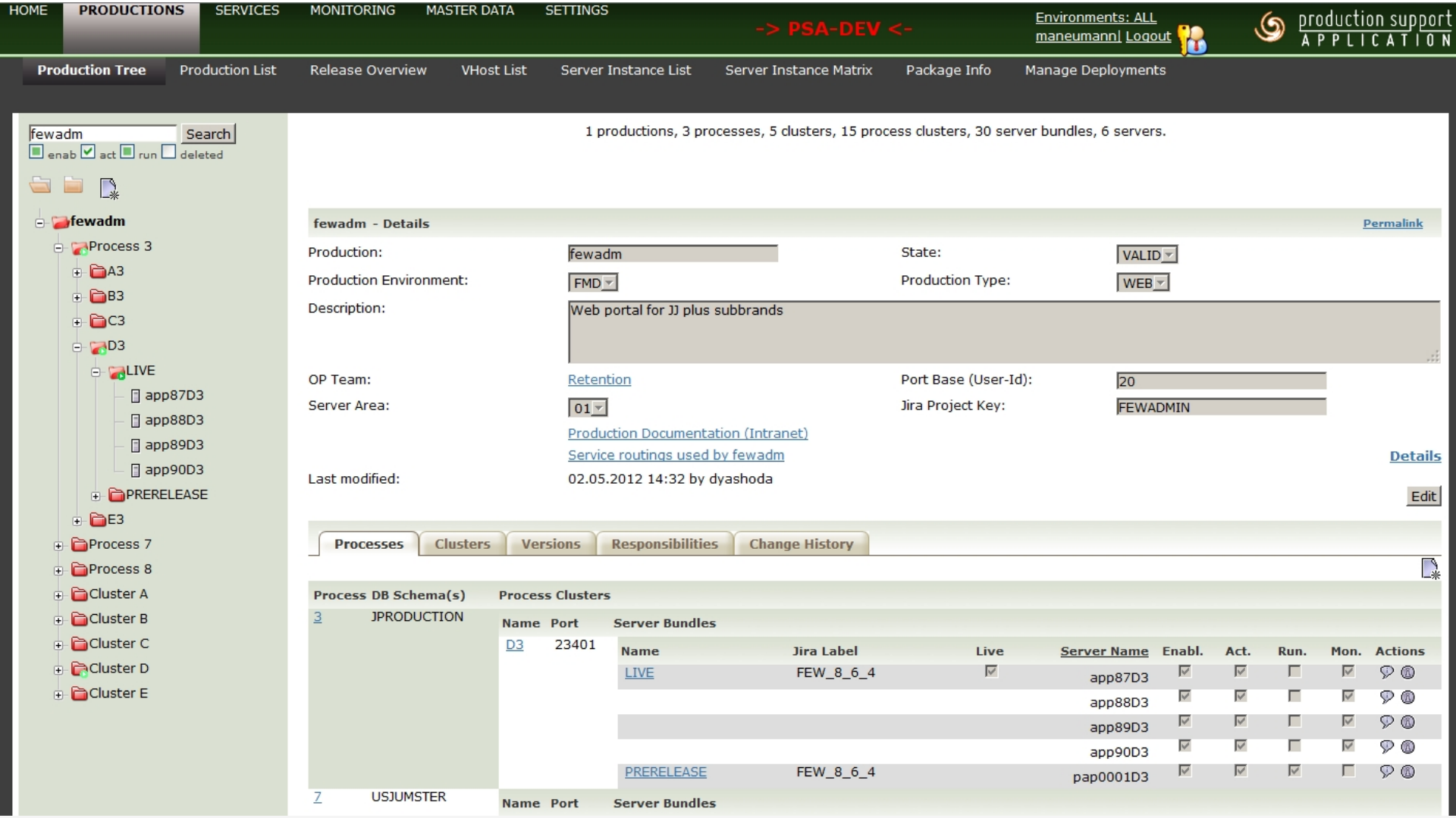
freenet digital

Implementation



- PSA frontend and backend functionality
- changeset feature to track every single configuration step
- assign monitoring configurations to different levels
  - production
  - process
  - server instance
- avoid duplication of configurations - use overwriting mechanisms
- easy to implement use cases common for most of the hosts
- special monitoring requirements can also be realized
  - change the threshold of a certain trigger on a certain machine for a certain time frame

PSA



freenet digital

Lessons Learnt So Far

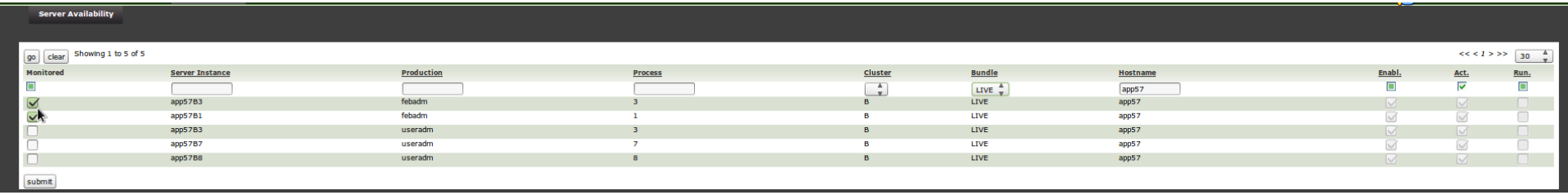
- started with template approach
  - always use templated items
  - more elegant, more transparent and safer
- turned out to be inflexible with regard to frequent changes
  - „changing only the delay of a single item“?
- users do not need to use the Zabbix frontend for making configurations
  - use a somehow „controllable“ environment where user errors can be foreseen
- complete integration is now done without templates
  - host items and triggers
  - gained a lot more flexibility
  - implementation delay due to switch

Host Availability Monitoring

- simple tcp checks on a dedicated agent targetted on http port („net.tcp.service.perf[http,app18.jamba.net,24831]“)

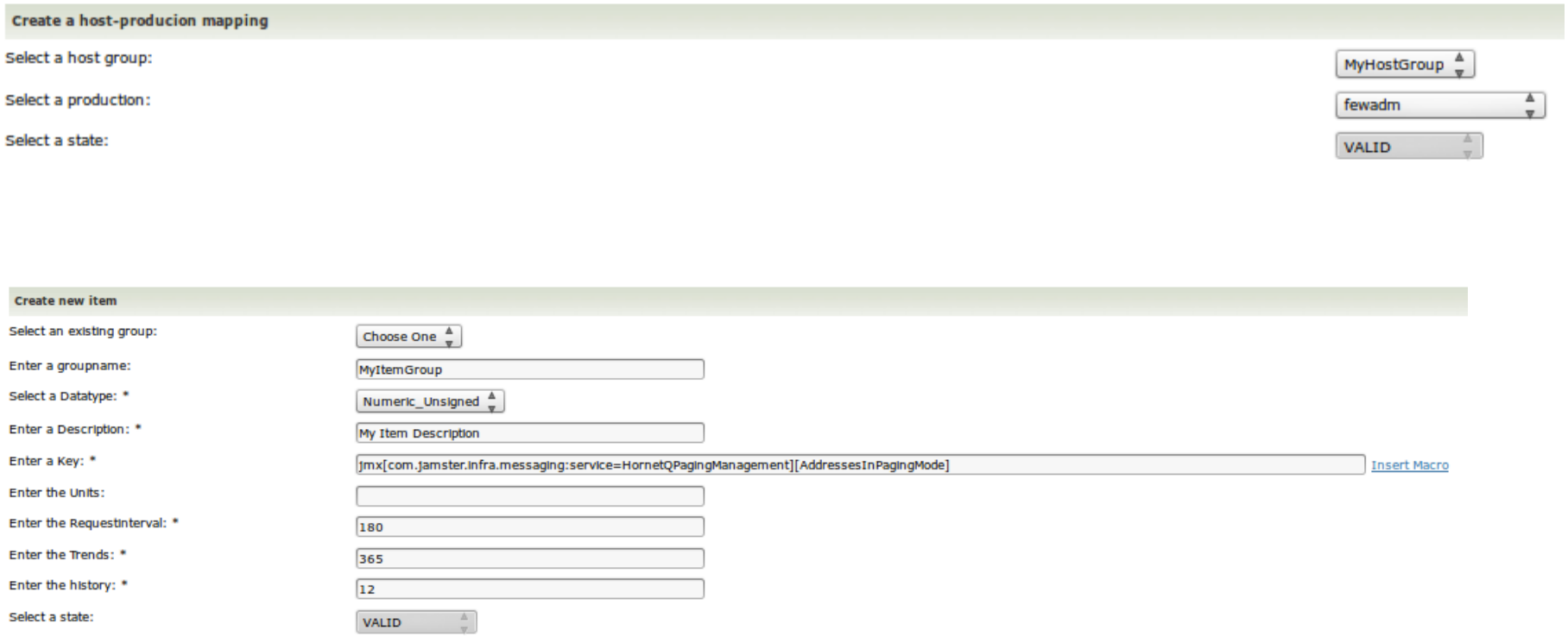
```
{"id":"0","method":"item.get","params":
  {"hostids":"300300000010450","search":
    {"description":"*LOSADMIN_app*76*_B3*Port*24831*",
     "name":"*LOSADMIN_app*76*_B3*Port*24831*"
    },
    "searchWildcardsEnabled":"1",
    "output":"shorten"
  }
}
```

```
{"jsonrpc":"2.0","result":[{"itemid":"300300000591350"}],"id":"0"}
```



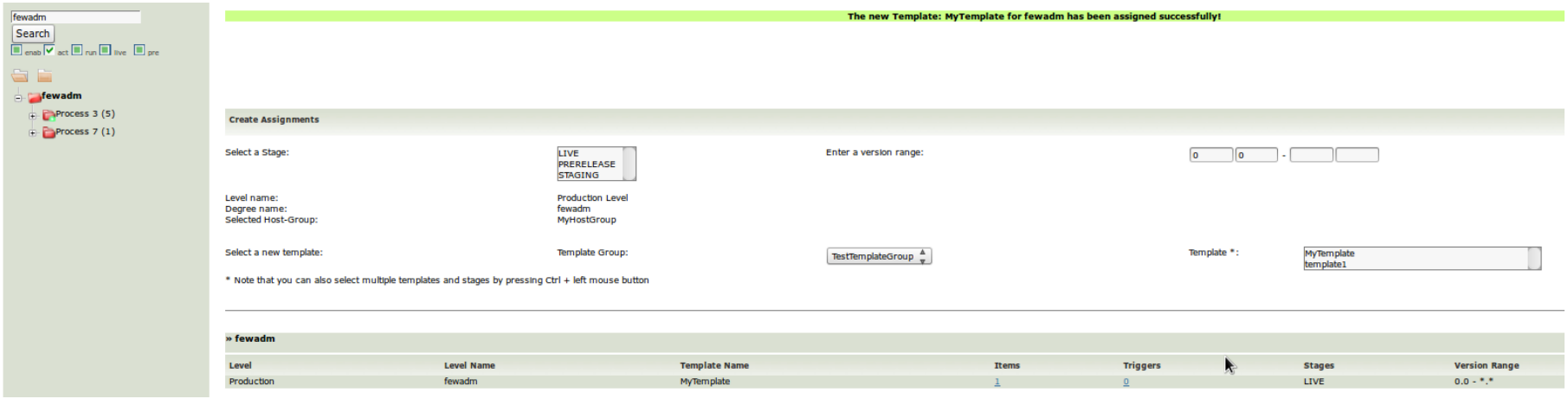
freenet digital

Create Hostgroups / Items / Triggers / Macros



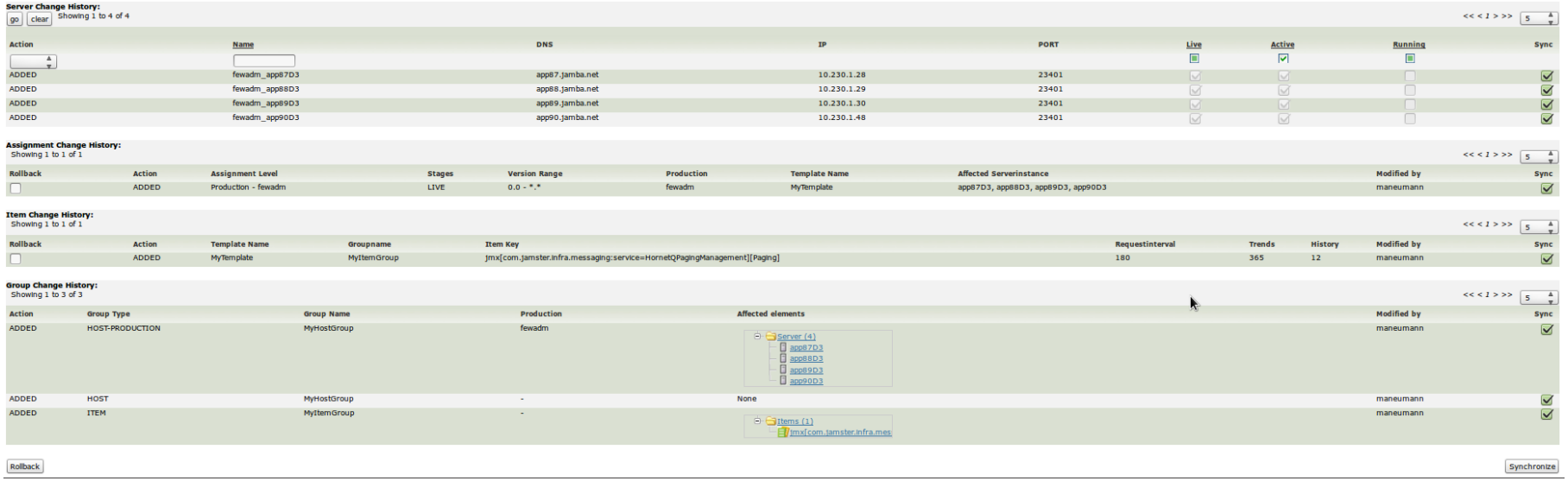
freenet digital

Assign Configurations



freenet digital

Synchronize with Zabbix



Server Change History:

go clear Showing 1 to 4 of 4 << < 1 > >> 5

Action	Name	DNS	IP	PORT	Live	Active	Running	Sync
ADDED	fewadm_app87D3	app87.jamba.net	10.230.1.28	23401	☑	☑	☐	☑
ADDED	fewadm_app88D3	app88.jamba.net	10.230.1.29	23401	☑	☑	☐	☑
ADDED	fewadm_app89D3	app89.jamba.net	10.230.1.30	23401	☑	☑	☐	☑
ADDED	fewadm_app90D3	app90.jamba.net	10.230.1.48	23401	☑	☑	☐	☑

Assignment Change History:

Showing 1 to 1 of 1 << < 1 > >> 5

Rollback	Action	Assignment Level	Stages	Version Range	Production	Template Name	Affected Serverinstance	Modified by	Sync
☐	ADDED	Production - fewadm	LIVE	0.0 - *.*	fewadm	MyTemplate	app87D3, app88D3, app89D3, app90D3	maneumann	☑

Item Change History:

Showing 1 to 1 of 1 << < 1 > >> 5

Rollback	Action	Template Name	Groupname	Item Key	Requestinterval	Trends	History	Modified by	Sync
☐	ADDED	MyTemplate	MyItemGroup	jmx[com.jamster.infra.messaging:service=HornetQPagingManagement][Paging]	180	365	12	maneumann	☑

Group Change History:

Showing 1 to 3 of 3 << < 1 > >> 5

Action	Group Type	Group Name	Production	Affected elements	Modified by	Sync
ADDED	HOST-PRODUCTION	MyHostGroup	fewadm	Server (4) app87D3 app88D3 app89D3 app90D3	maneumann	☑
ADDED	HOST	MyHostGroup	-	None	maneumann	☑
ADDED	ITEM	MyItemGroup	-	Items (1) jmx[com.jamster.infra.mes	maneumann	☑

Rollback Synchronize

freenet digital

Performance Lessons Zabbix API (1.8)

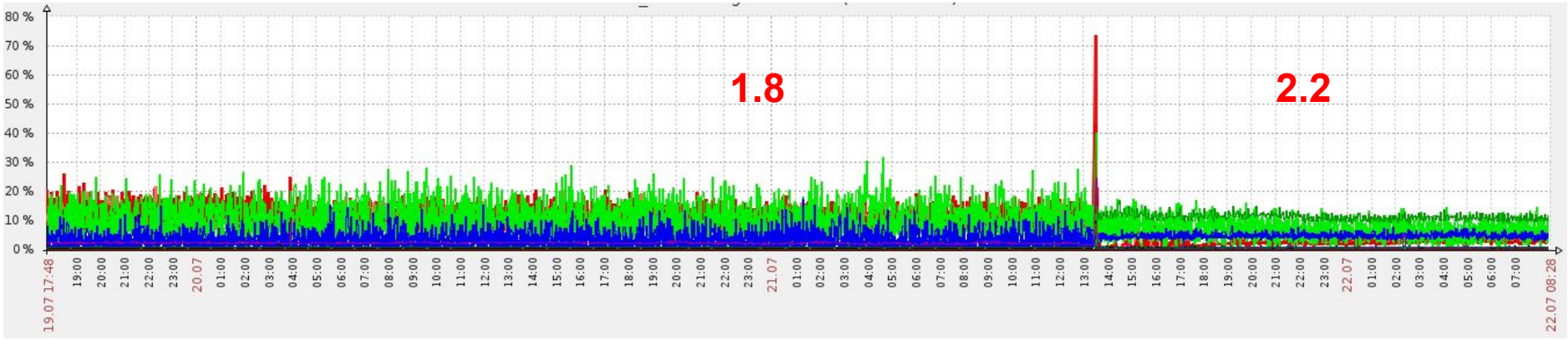
- use bulk operations where applicable
  - hostgroup.massAdd, hostgroup.massRemove
  - usermacro.massAdd, usermacro.massUpdate, usermacro.massRemove
  - item bulk methods not supported
    - one item.create = one remote call
- insufficient error handling („Invalid params“)
  - more round-trips required for querying the Zabbix state („is the item really existing?“)
  - clean error handling would be required in api client to provide suitable exceptions to the applications embedding the client
- additional calls required due to internal representation of Zabbix objects
  - trigger.get() response exposes function ids in its expression

Zabbix Upgrade to 2.2

- production was upgraded in July
- clone database and execute test migration (on real data!)
  - 1.8 → 2.0 (took 8,5h)
  - 2.0 → 2.2
- provide enough temp tablespace for MySQL!
- high effort to make templates compatible to 2.0
  - „interesting“ xml structures
- refactored the availability checks
  - move from template to host (thanks to the interface concept introduced in 2.0)
- merge of production and integration planned
  - maintain only one instance

Zabbix Upgrade to 2.2

- History syncer usage decreased by ~5%



- faster frontend
- interface concept allows to have more than one agent on a host
  - required for Java + Native Zabbix agent
- JMX Agent not tested yet
  - very likely too slow due to JMX remote calls

freenet digital

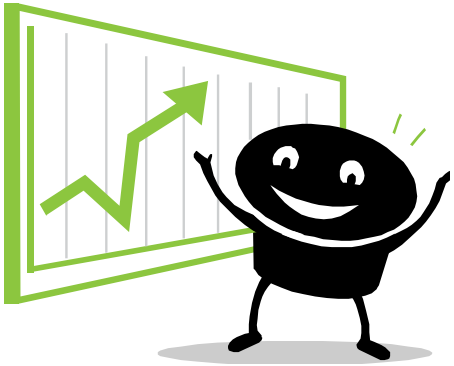
Performance Lessons of 6 Zabbix years operations

- MySQL database tuning required for high tx throughput
  - InnoDB parameters (log file size, query cache)
- MySQL partitioning to avoid Zabbix internal housekeeper
  - avoid concurrent write blocking behavior
  - required for big transactional tables like history*, event, trends*
- Database: VM - Hardware - VM
  - database is running on a VM backed by fast storage since May
- use internal metrics for analysis
  - current load of history syncer processes
- visualize as much as possible to gain attention
- do not delay version updates for too long

Performance Lessons of 6 Zabbix years operations

Number of hosts (monitored/not monitored/templates)	1007	269 / 234 / 504
Number of items (monitored/disabled/not supported)	43720	39930 / 372 / 3418
Number of triggers (enabled/disabled) [problem/ok]	44647	40711 / 3936 [159 / 40552]
Number of users (online)	38	5
Required server performance, new values per second	270.28	-

- room for more items and hosts



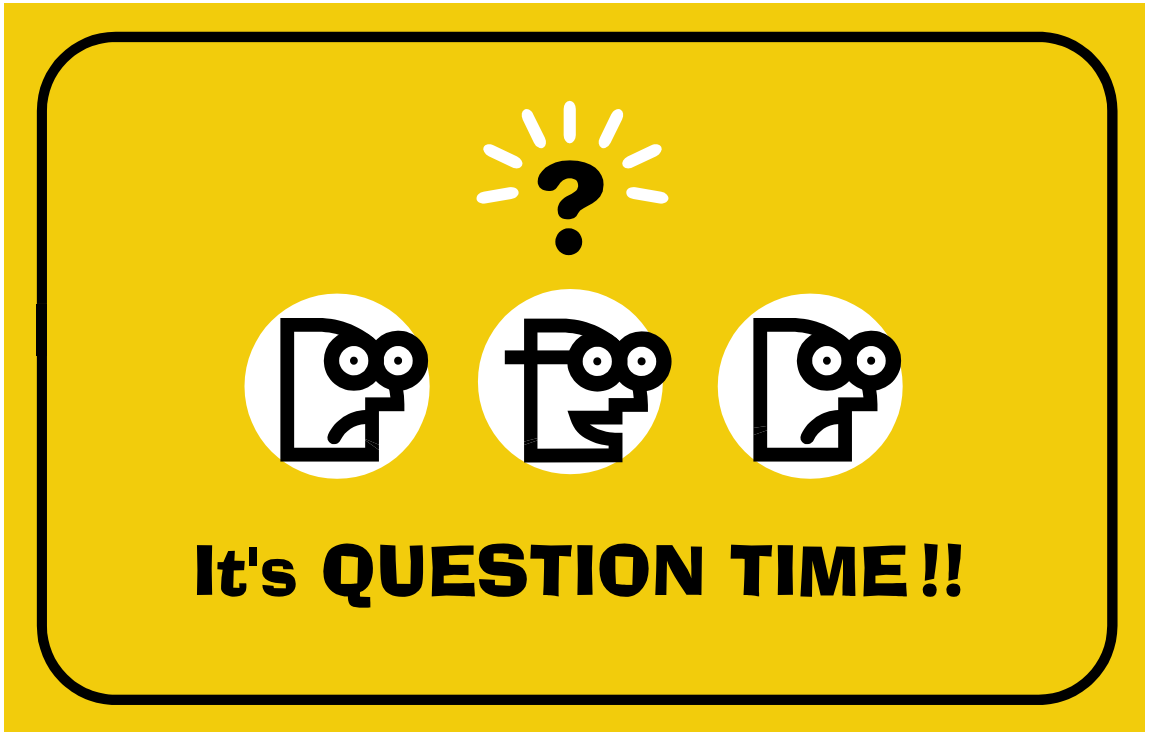
Monitoring Automation Project Outcome

- started the project in 2011 (…and still not finished)
- a lot of lessons learnt during that time
- complex business requirements
- major refactorings (template → host) and rewrites
- only small business priority
  - student employees are working full-time on it
  - monitoring is only small part of S&A‘s tasks
- additional efforts for Zabbix 2.2 upgrade
- live switch targetted for Jan 2015

Summary

- decision for Zabbix was a good one **definitely**
- high performance and scalabilty is possible - but not of-of-the-box!
- professional support is recommended for mission-critical systems
  - esp. when monitoring adminstration is not the only task you have
  - honor open source efforts done by Zabbix S.I.A.
- automation (using the api) is not as simple as we thought
  - knowledge about Zabbix internals was required
  - api and its documentation evolved over the time…
- tool acceptance is important
- never stop learning
- report errors and features requests back to the community
- attend Zabbix conference ☺

Q & A



freenet digital