



YADE

Architecture and Mode of Operation

Open Source Users and Commercial Customers

- Open Source Users in more than 100 countries
- 94 commercial customers in 20 countries



Contents

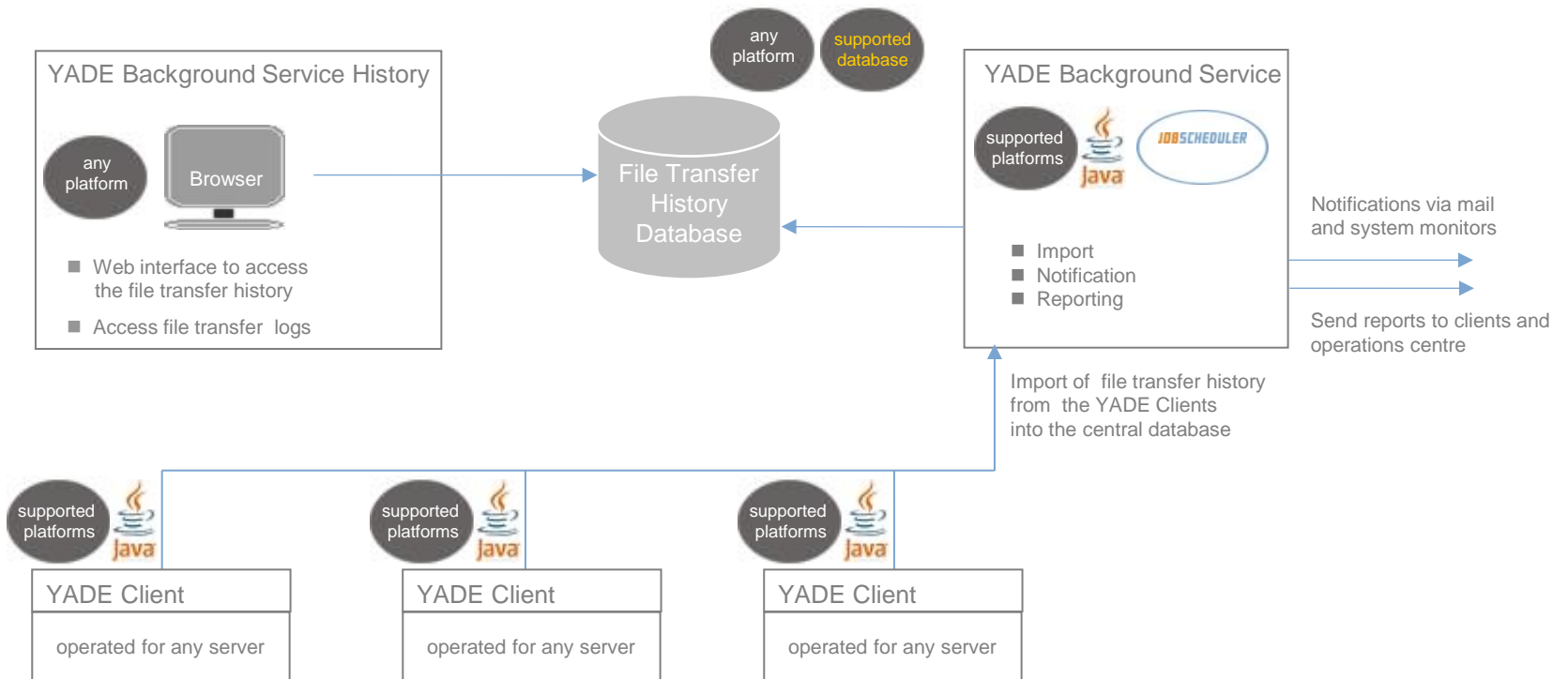
- Architecture and Components
 - Component Architecture
 - Implementation Architecture
- YADE Client
 - YADE Command Line Interface
 - YADE API
 - YADE JITL Job
- YADE Background Service
 - YADE Background Service Jobs
 - YADE Background Service History Viewer
- Mode of Operations

Backup

- Gaps and Solutions
 - Gaps of individual file transfer scripts and benefits using YADE
 - Logging, Reporting, Automation

- Architecture
- Components
- Supported Platforms

YADE Client and Background Service Components



- YADE Client
 - YADE Command Line Interface
 - Command line utility for integration with batch files
 - YADE API
 - Enables applications to use the YADE functionality.
 - YADE JITL Jobs
 - Seamless integration with JobScheduler for automated transfer
- YADE Background Service
 - YADE Background Service Jobs
 - Imports the transfer history of the YADE Clients into a database
 - Provides notifications in case of errors
 - YADE Background Service History
 - GUI for monitoring file transfers
 - Allows to search and browse the file transfer history
 - YADE Background Service Reporting
 - Generates custom reports for file transfers
 - Mails reports of effected transfers

- Supported platforms of YADE Client
 - YADE Client can run on all operating systems that support a Java Virtual Machine.

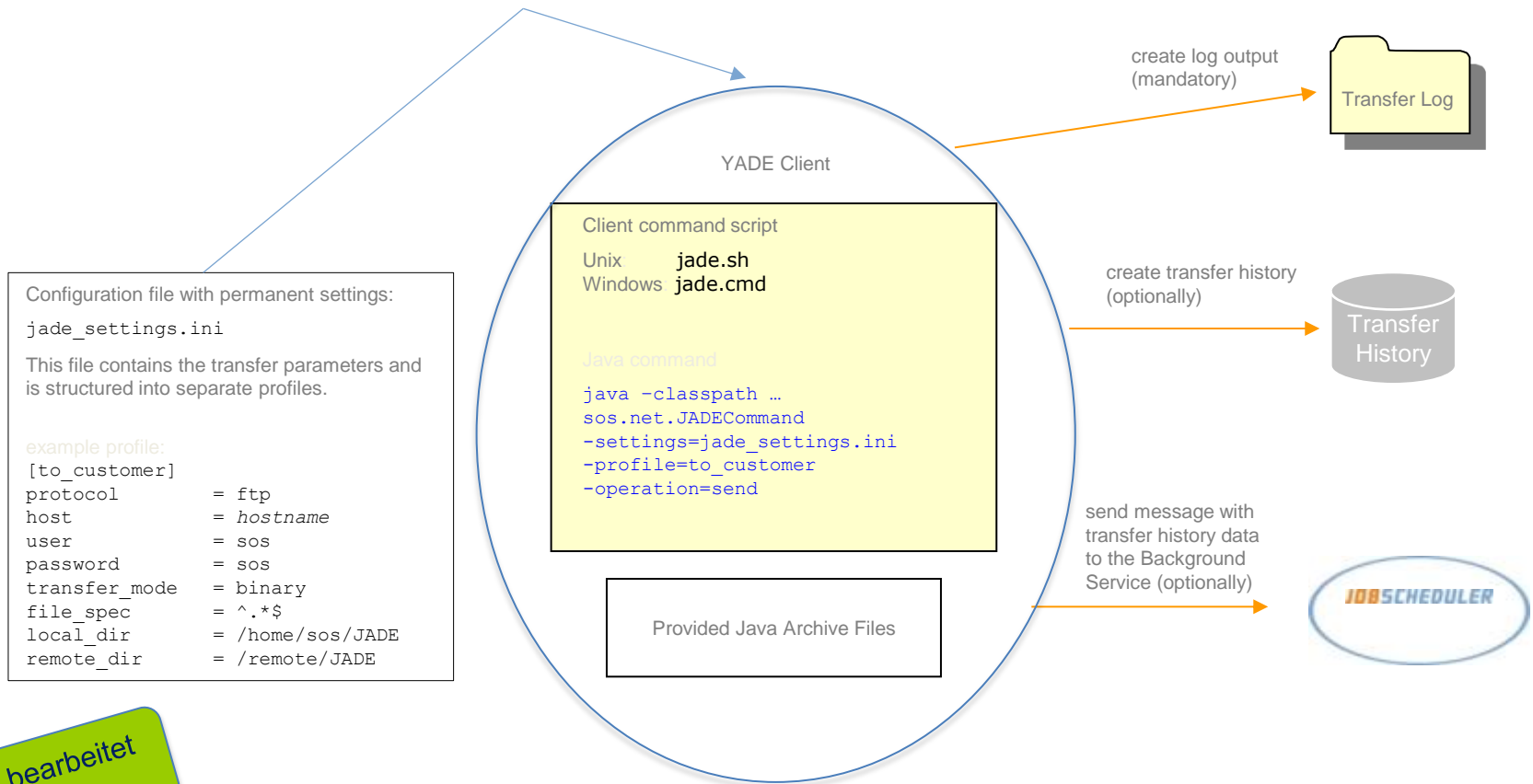
- Supported platforms of YADE Background Service
 - Service is integrated with JobScheduler and uses the same platforms
 - Operating Systems
 - Microsoft Windows: 2008, 2012, Vista, 7, 8, 10, ...
 - Linux: RedHat, CentOS, ...
 - Supported Databases
 - DB2, Oracle, SQL Server, Sybase
 - MySQL, MariaDB, PostgreSQL

- Supported platforms of YADE Background Service History
 - GUI is operated with any servlet container
 - Jetty (provided)
 - Tomcat

- YADE Client Usage
- YADE Client Prerequisites
- YADE Client Basic Features
- YADE Client Advanced Features
- Basic File Transfer
- Server to Server Transfer

Usage example of the YADE Client Command Line Interface:

```
jade.sh -settings=ftp_settings.ini -profile=to_customer -operation=send
```



Muss bearbeitet werden

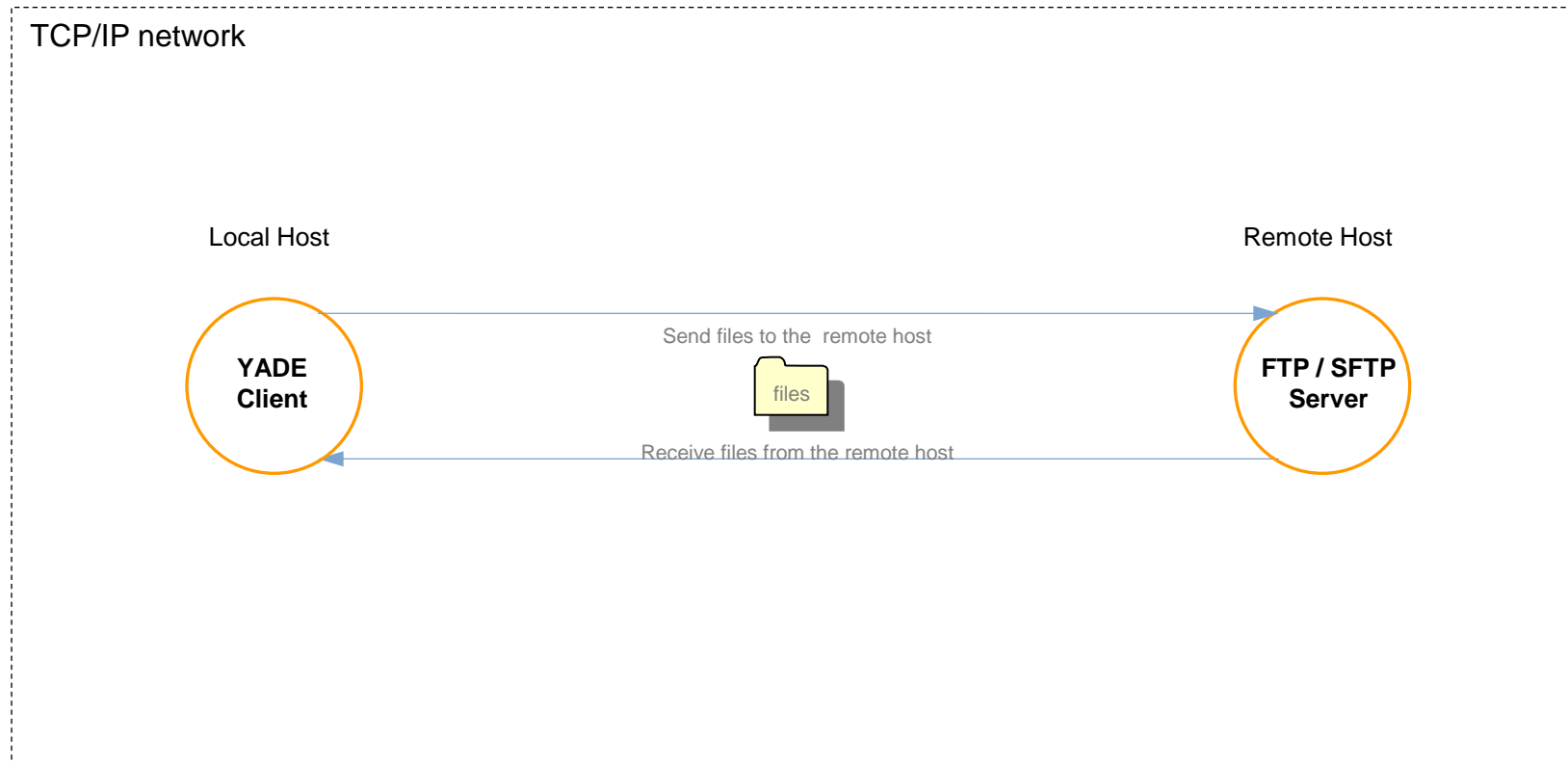
- System Requirements
 - Java Runtime Environment (JRE) starting from version 1.8
 - The YADE Client can be used standalone
 - Use of additional YADE Background Service components is optional

- Basic Features
 - Send and receive files by FTP to/from some target host.
 - Send and receive files by SFTP to/from some target host.
 - Execute commands by SSH.
 - Password and public/private key authentication are supported for SFTP.
 - The parameterization is effected by command line parameters and/or by configuration files.
 - All parameters are specified on the local host exclusively, this applies in the same way when using a jump host as local parameters are dynamically forwarded to the jump host.
 - Logging and error handling are provided, errors are detected on a per file basis.

- Advanced Features
 - Transfer of partially matched files
 - Recursive transfer of files in directories
 - Atomic Transfer
 - Transactional Behavior
 - Check transferred files for completeness (data integrity checking)
 - Handling of empty files
 - Renaming of files
 - Use of alternative access data and credentials
 - Support for UNC paths in target directories
 - Track record of file transfers
 - Configurable log structure and debug information

Components YADE Client for File Transfer

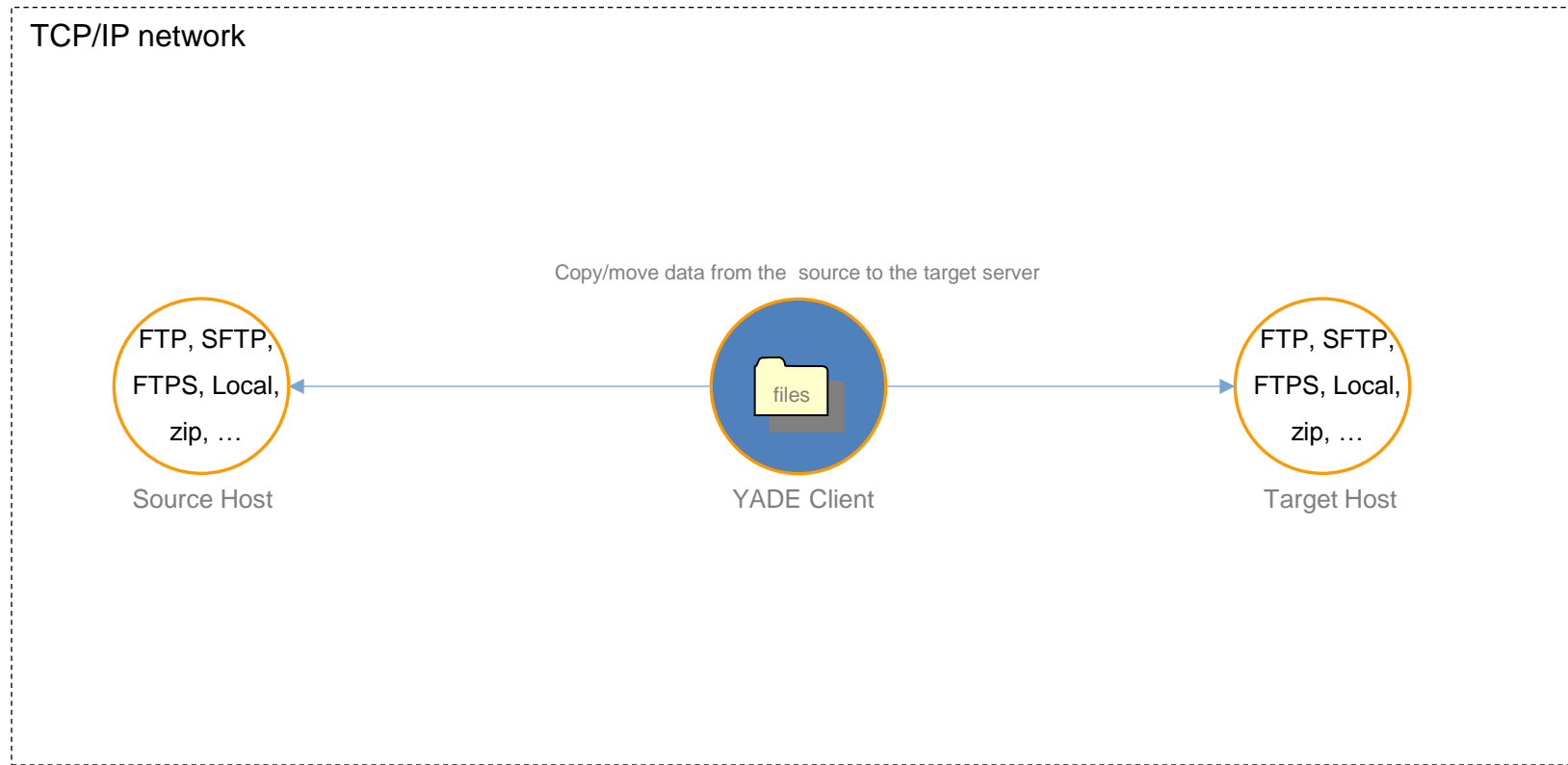
Standard File Transfer: send and receive



Muss bearbeitet
werden

Components YADE Client

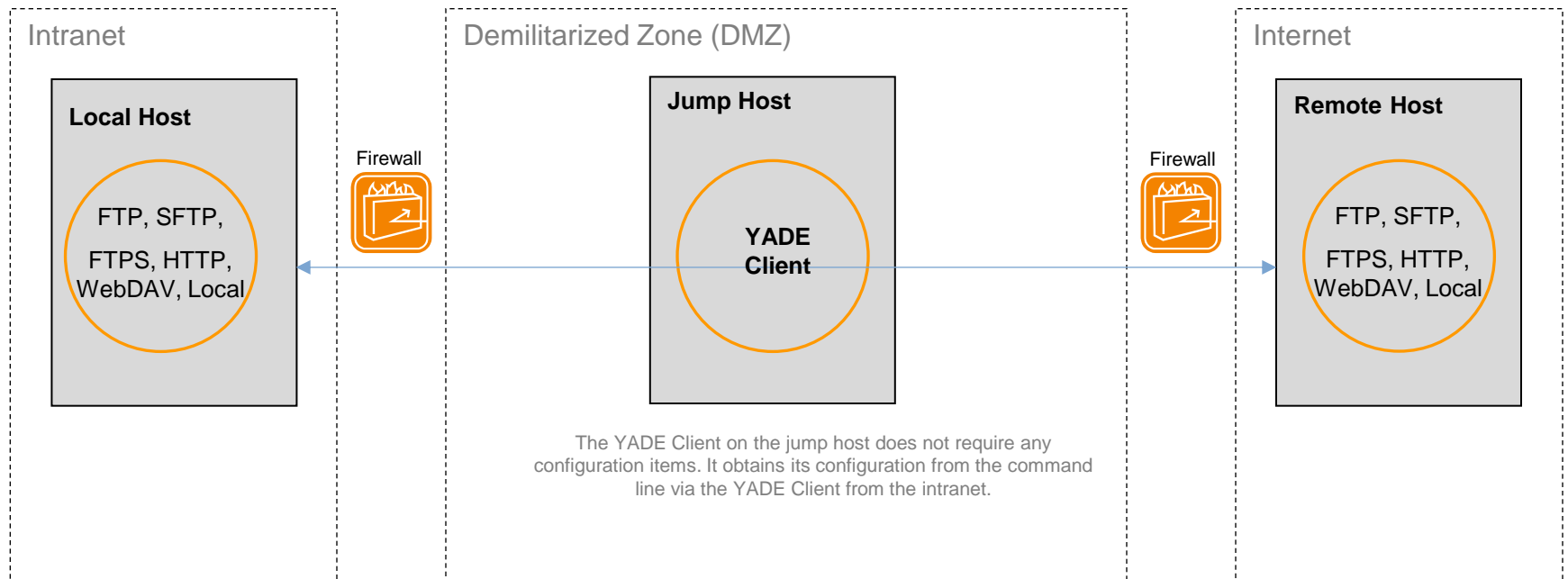
Server to Server: copy and move



Muss bearbeitet werden

YADE Client with Jump Hosts

File Transfer via an intermediate DMZ: transfer files from and to locations in the Internet



- Why use a Jump Host?
 - A jump host is located in the DMZ, therefore it can access servers at customer sites via the internet. Local servers quite often are excluded from internet access for security concerns.
 - Jump hosts are used in order to transfer files without a touch down to the target server which is either an intranet server or a server in the internet, depending on the transfer direction.

- Security Considerations
 - When operating YADE for a jump host then no configuration data will be stored on that host. Instead, the YADE Client on the jump can get all parameters by command line parameters. No additional YADE Client instance. Is needed.
 - Files are never stored on the jump host. Therefore they must not be removed on completion of the transfer.
 - No log output is created on the jump host, therefore no private data are stored on that host.

- Architecture and Components
- File Transfer History
- Web GUI
- Reporting

- Connectivity
 - YADE Clients can be configured to use an UDP or TCP port for communication with the YADE Background Service.
 - YADE Background Service can access YADE Client hosts by YADE JITL Job to collect History Files.
- Automation
 - Open Source JobScheduler (<http://www.sos-berlin.com/jobscheduler>) for process automation
 - Java Runtime Environment starting from version 1.8
 - Any supported database in order to store the file transfer history

- Import
 - YADE import of the file transfer history that is created by YADE Clients into a central database.
 - Immediate import is effected by YADE Clients via UDP or TCP.
 - Fail-safe import by repeated execution in case of connectivity issues between clients and Background Service.
 - Compliant transfer history of all incoming and outgoing file transfers.
 - Integration with the Open Source JobScheduler (<http://www.sos-berlin.com/jobscheduler>) for workflow automation.
- Alerting
 - Notifications on errors and warnings in any file transfers are sent by mail.
 - Network monitoring solutions can be integrated based on queries to the file transfer history database.

- Architecture and Components
- Prerequisites and Features
- Sample Screenshot

- Prerequisites
 - Web server (Apache, IIS etc.) with PHP starting from version 5.x.
 - PHP database extension for file transfer history database.
 - YADE Background Service component that feeds the file transfer history database.
- Features
 - Centralized view of each and every YADE file transfer operation in the system.
 - Monitoring of incoming and outgoing transfers.
 - A traffic light signals errors and warnings of current transfers.
 - Detailed information on each transfer as hosts, direction, files and error messages.
 - Search options for the file transfer history
 - Grouping of file transfers per date, host, file, result etc.



Muss bearbeitet werden

Web GUI: Sample Screenshot

mandator target file name

state source file name

operation target host

protocol source host

timestamp to

only last state refresh interval (all 60 s)

<< >> 1 2 3 4 5 6 7 8 9 10 11 hits : 157 (1 to 15)

show details

Search Criteria

Navigation Bar

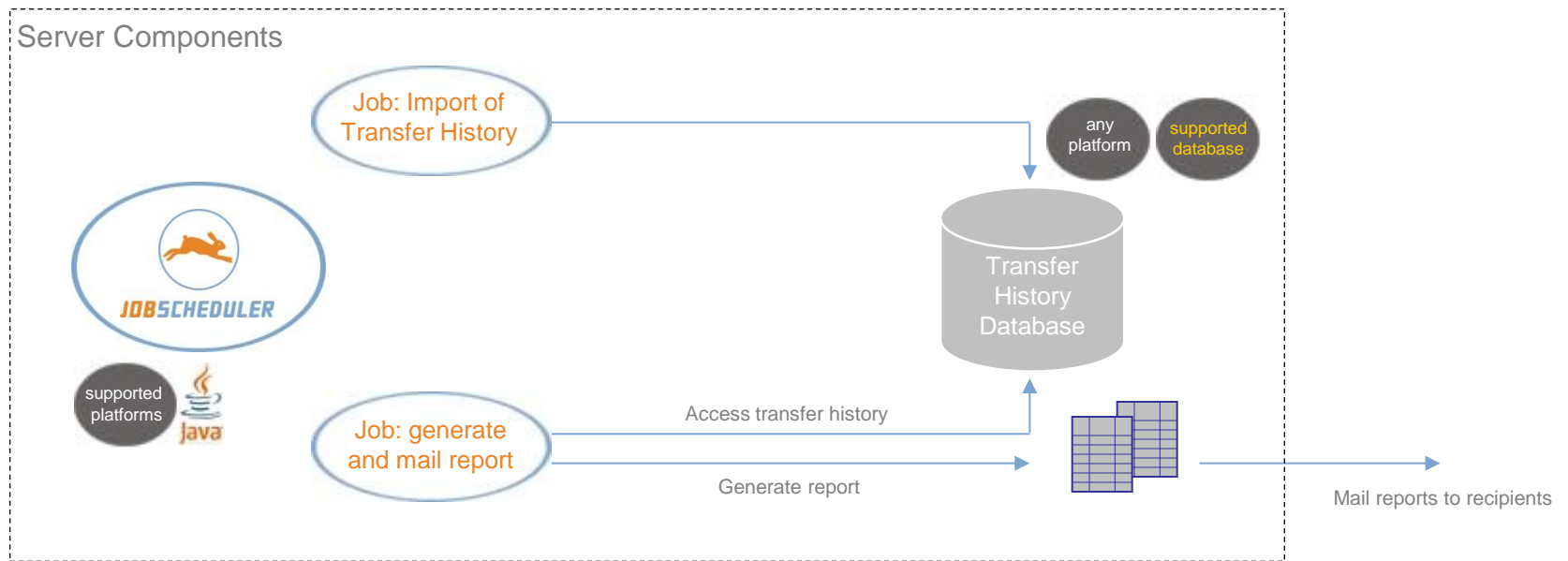
Detailed Transfer History

	mandator ↓↑	timestamp ↓↑	operation ↓↑	protocol ↓↑	target file name ↓↑	file size ↓↑	source host ↓↑	target host ↓↑
■	sos	07/20/2009 15:32:46	receive	ftp	a.xml	818	8of9	wilma
ftp processing failed: could not process file transfer: file transfer failed: Could not open stream for a.xml. Perhaps the file does not exist. Reply from ftp server: 550 a.xml: Das System kann die angegebene Datei nicht finden. .								
■	sos	07/20/2009 15:32:45	send	ftp	unknown	0	wilma	8of9
ftp processing failed: could not process file transfer: error in sos.net.sosftp.SOSFTPCommandSend.sendFile could not send files, cause: java.lang.Exception: .. file [/tmp/inbound/1/a.xml] does not exist								
■	sos	07/20/2009 15:32:43	receive	sftp	DD.XML	495	wilma	wilma
■	sos	07/20/2009 15:32:43	receive	sftp	KOPIE VON SOSFTP.XML	136088	wilma	wilma
■	sos	07/20/2009 15:32:42	receive	sftp	COMMONS-NET-1.2.2.JAR	153280	wilma	wilma
■	sos	07/20/2009 15:32:42	receive	sftp	a.xmlXX	495	wilma	wilma
■	sos	07/20/2009 15:32:41	receive	sftp	SOS.SETTINGS.JAR	88109	wilma	wilma
■	sos	07/20/2009 15:32:41	receive	sftp	jobeditor.sh	0	wilma	wilma
■	sos	07/20/2009 15:32:41	receive	sftp	trilead-ssh2-build211.jar	248820	wilma	wilma
■	sos	07/20/2009 15:32:40	receive	sftp	Mmm.xml	136479	wilma	wilma
■	sos	07/20/2009 15:32:39	receive	sftp	SOSFTP.XML	136088	wilma	wilma
■	sos	07/20/2009 15:32:39	receive	sftp	SOSFTP.XSL	87208	wilma	wilma
■	sos	07/20/2009 15:32:38	receive	sftp	SOSFTP.CMD	588	wilma	wilma
■	sos	07/20/2009 15:32:38	receive	sftp	scheduler~~	6595201	wilma	wilma
■	sos	07/20/2009 15:32:14	receive	sftp	1a.xml	495	wilma	wilma

Muss bearbeitet
werden

- Creating and E-Mailing Reports
- Reporting Features
- Example Report

- Report Design
 - Reports can be designed for multiple formats such as HTML, PDF, XLS etc.
 - Reports can be customized at your convenience. This includes the layout, the level of detail and the period that is covered by the report.
 - Reports are based on three components: a database query, a report layout and a report job.
 - All report handling is scheduled by JobScheduler.
 - Default reports are provided for a daily and monthly overview of file transfers.
- Report Handling
 - Reports can be stored to disk.
 - Reports can be sent by E-Mail.



- Reports are created from the transfer history database
- Report generation is implemented as job in the JobScheduler
- Run time components for report generation are provided as Java classes
- The Design time component for reports is provided as Java desktop application

Muss bearbeitet werden

- Report Generation
 - For report generation a template job is available that includes Open Source components with compatible licenses.
 - Reports are implemented with the Open Source software JasperReports (<http://www.jasperforge.org>) that provides the runtime environment and the design component.
- Report Automation
 - Processing of reports is scheduled by the Open Source JobScheduler (<http://www.sos-berlin.com/jobscheduler>)
 - JobScheduler provides time events for report processing on a per job basis. Report jobs can handle different parameter sets depending on the scope and target audience of a report.

FTP File Transfer History Report for Month December

Mandator: test

Date	OP	Source Host	Target Host	Prot.	Local Directory	Target Directory	Filename	Size	Status	Error Messages
05-Dez-08 10:55:22	receive	hp06b	hp06b.echbruedom.local	ftp	/app/int/sosedu/tmp/inbound	/app/int/sosedu/tmp/client/outbound	job.xml	808	success	
05-Dez-08 10:55:22	receive	hp06b	hp06b.echbruedom.local	ftp	/app/int/sosedu/tmp/inbound	/app/int/sosedu/tmp/client/outbound	2.txt	6	success	
05-Dez-08 10:55:22	receive	hp06b	hp06b.echbruedom.local	ftp	/app/int/sosedu/tmp/inbound	/app/int/sosedu/tmp/client/outbound	1.txt	6	success	
05-Dez-08 10:55:22	receive	hp06b	hp06b.echbruedom.local	ftp	/app/int/sosedu/tmp/inbound	/app/int/sosedu/tmp/client/outbound	order.xml	892	success	

Customizable level of detail

Reports filtered for specific customers

Applicable report period

Muss bearbeitet werden

■ Scripts

- The implementation with individual file transfer shell scripts is quite often platform-dependent and error-prone, this means most of the time there is no sufficient proof that a file transfer has definitely been successful.
- In most cases there is no error handling, e.g. no repeating of the file transfer in case of failure.
- Implementing with individual shell scripts means that it is required for the configuration to be organized individually. The change of a protocol, e.g. from FTP to SFTP requires a new implementation each time.
- No data integrity checking after successful file transfer.
- No support for advanced regular expressions to select files for transfer.

■ YADE

- Using a standard implementation (Java) for all server platforms with no additional requirements for the installation.
- Error detection and error handling.
- Support of protocols FTP, FTP/S, SFTP (public / private key).
- Batch processing via command line and configuration file.
- YADE provides inherent mechanisms to ensure data integrity checking through hashing (md5).
- YADE supports advanced regular expressions for file selection and for renaming operations.

■ Scripts

- Hard to maintain host/source credentials or other parameters required for file transfer.
- No implicit mechanism to ensure successful transfer of a group of files.
- No implicit mechanism to specify secondary host/login credentials should an FTP server not be accessible or in case of any other FTP login related errors.
- Development/Testing/Debugging of new/changed FTP script takes a lot of time, no way to ensure everything will work in production other than actually transferring files.

■ YADE

- All the settings required for file transfer i.e. host name/username etc. can be grouped in one profile and stored in YADE Client's configuration file.
- One profile can be used for send/receive by multiple file transfer operations.
- Ensure atomic transfer i.e. the transfer will only be considered successful if all the files in a group are successfully transferred, in the event of failure in transfer of any one of the files from a group the whole transfer will be "rolled back" and will be considered as unsuccessful.
- YADE's test mode provides ease of testing/checking the entire file transfer operation i.e. login credentials and directories or file names matching with regular expressions etc. without actually transferring any files.

■ Scripts

- Protocols are not standardized
- Protocol files are not intuitively understood.
- Protocols of shell scripts do not show an error state explicitly.
- Conventional FTP batch script's error messages don't point at the exact cause of the error and often some error message doesn't break the flow of the batch script, resulting in undetected unsuccessful file transfers.
- Hard to comprehend/debug logs generated by FTP scripts, often FTP log messages are written together with batch job's log.

■ YADE

- Protocols are standardized with a structured and configurable format.
- A file transfer history for all transfers is created and optionally stored in a central file transfer history in text files or databases.
- Automatic alerting in case of errors.
- A file transfer history for all transfers is created and optionally stored in a central file transfer history in text files or databases.
- Configurable debug log levels provide varied levels of information about file transfer.

■ Scripts

- No information about received or sent files is centrally available.
- No analysis is possible concerning the error occurrence of transfers that have been undertaken with shell scripts.
- No compliance reporting is available, e.g. a daily or weekly reports that show frequency and date of the transferred files.

■ YADE

- A centrally available GUI is used in order to monitor and to analyse the file transfers. The Web GUI allows to search the complete file transfer history.
- Automatic generation and mailing of file transfer history reports, configurable KPI reports.
- Individual reports can be created with a standard report generator by querying the transfer history database.
- Highly customizable reporting capability (using other open source tools) provides enhanced IT auditing and service level compliance auditing.
- Greater vigilance on critical enterprise data exchange through file transfer operations.
- Different transfer history reports can be produced from the same data, i.e. reports for Customers, Business Analysis, Administrators, IT Managers etc.

■ Scripts

- Conventional FTP scripts can be scheduled with cron tab/OS's scheduler/third party schedulers.
- Only time based execution of FTP scripts is possible with cron tab/OS's native scheduler.
- Conventional FTP scripts lack in event based and more advanced file handling operation i.e. start of FTP operation on arrival of specific files in a target directory.
- Hard to program event based /conditional file transfer operation in shell scripts.

■ YADE

- Seamless integration of YADE with Open Source JobScheduler provides End-to-End automation of file transfer operation and file processing. (YADE and Open Source JobScheduler are provided by SOS GmbH).
- Event/time based pre/post file transfer processing.
- Event/time driven file transfer operation.
- Event/time based polling of server/directories for specific files (parameterised polling time and interval)
- Directory monitoring for files matching with regular expressions.
- Efficient 24x7 fully automated file transfer and processing.
- Easy integration with existing IT infrastructure.

Questions? Feedback?



Software- und
Organisations-
Service GmbH
Giesebrechtstr. 15
D-10629 Berlin
Germany

[mailto: sales@sos-berlin.com](mailto:sales@sos-berlin.com)
<http://www.sos-berlin.com>

