QlikView Publisher: Scale with Simplicity

QlikView

Today's challenge: BI software must support information flow

- Access to information in multiple sources

- Visible processes

- Single point of control

QlikView Publisher

Manage scheduling, distribution, and security for all
your QlikView applications and reports through a
central Web-based management console

QlikView Publisher—about getting things right

- The right information

- To the right user

- At the right time

- In the right format
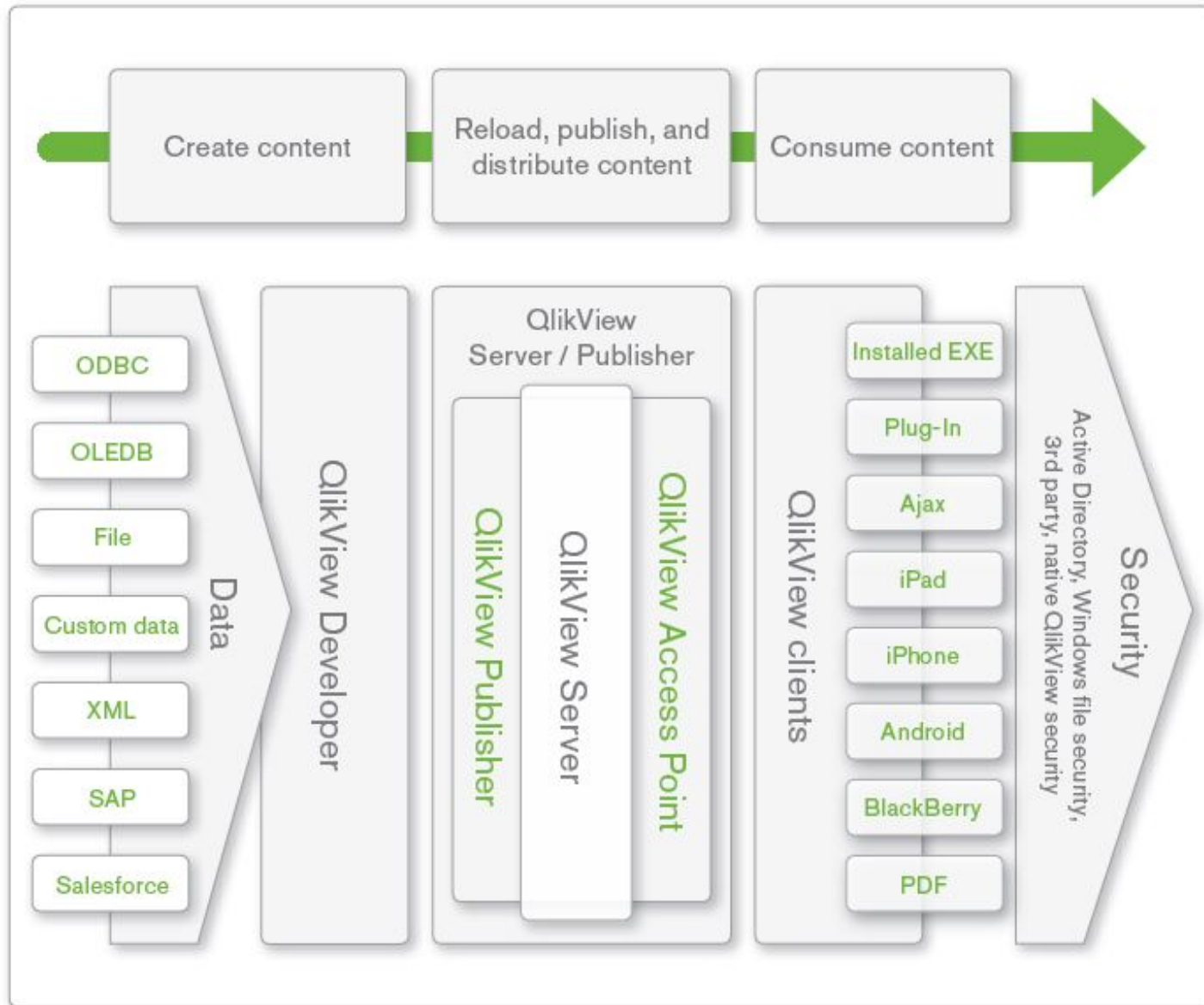
Benefits

- Timely, accurate information delivery

- Secure access to business data and analytics

- Delegated administration

- Enterprise-wide report distribution

- Scalability and manageability
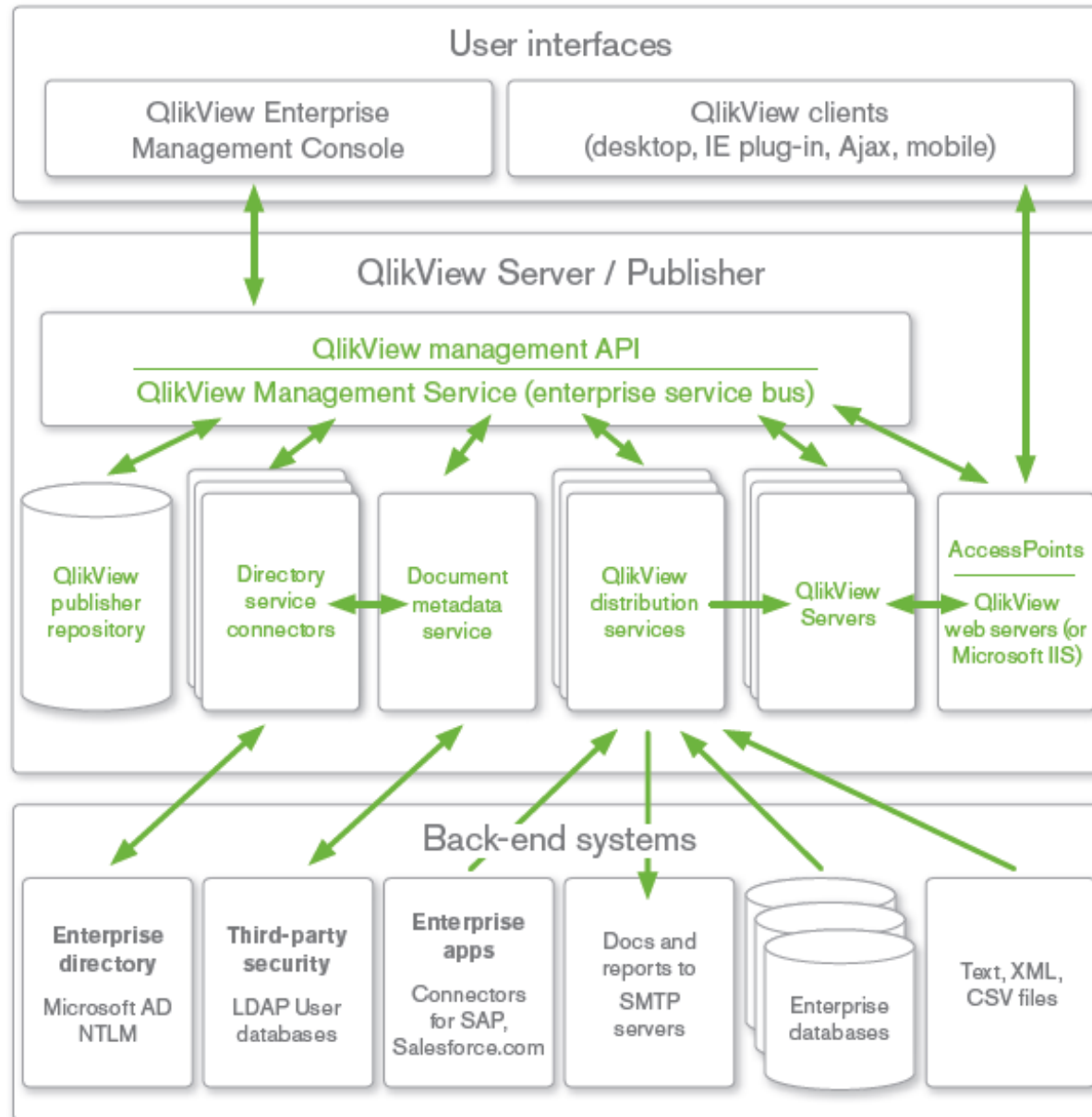
QlikView Publisher in context

QlikView Publisher and QlikView Server go hand in hand

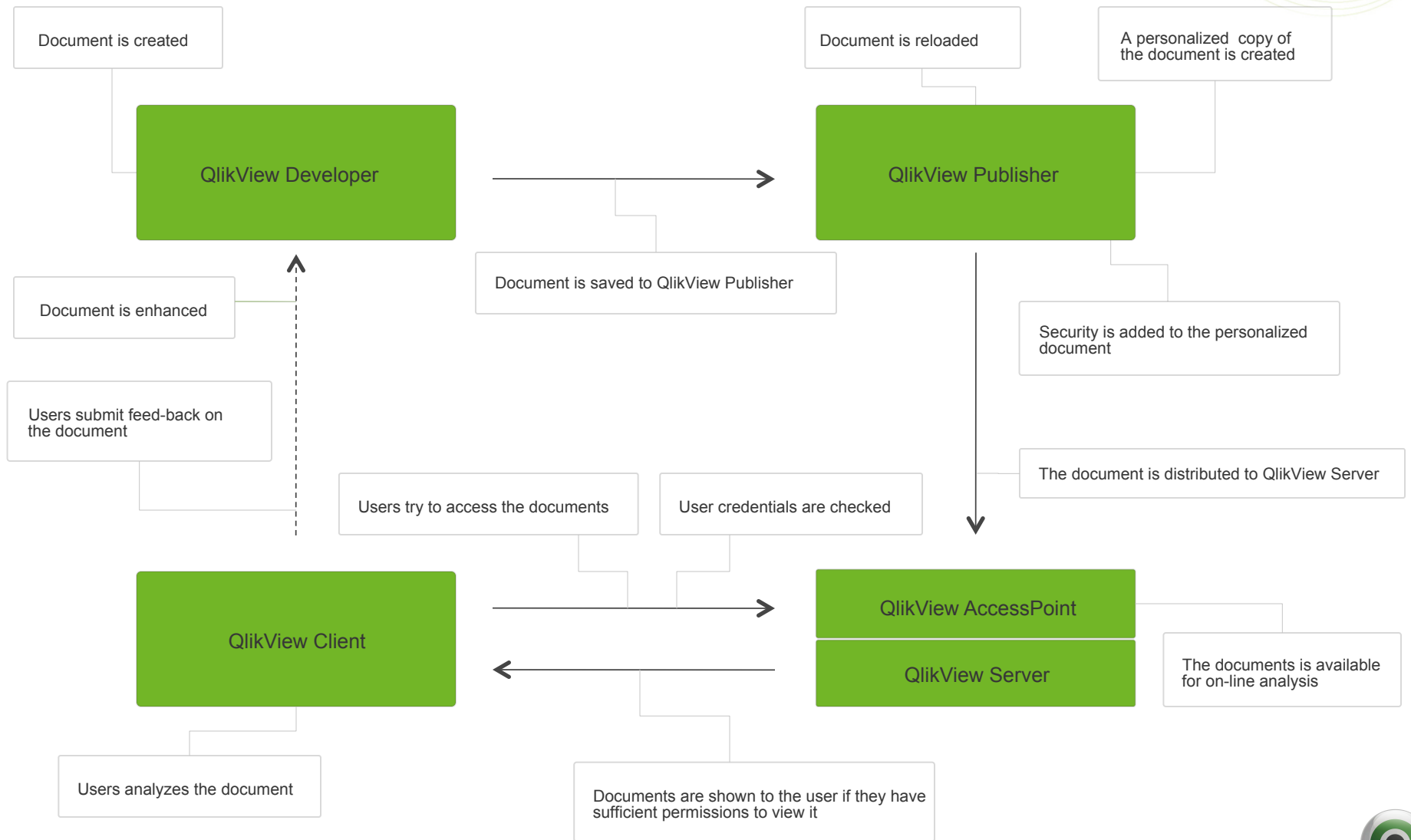| QlikView Publisher	QlikView Server
Publish and distribute QlikView files	Open QlikView documents and publish the contained data
Create subsets of QlikView documents (e.g., per user, per region, etc.)	Enable users to connect to a QlikView document and perform analysis
Share load across different parts of QlikView Publisher	Provide shared memory architecture for efficient QlikView usage
Facilitate management of large number of QlikView documents	Facilitate management of large number of users

QlikView Server / Publisher architecture

QlikView Publisher Workflow

Document is created

QlikView Developer

Document is enhanced

Users submit feed-back on the document

Document is saved to QlikView Publisher

Document is reloaded

A personalized copy of the document is created

QlikView Publisher

Security is added to the personalized document

The document is distributed to QlikView Server

Users try to access the documents

User credentials are checked

QlikView Client

QlikView AccessPoint

QlikView Server

The documents is available for on-line analysis

Users analyzes the document

Documents are shown to the user if they have sufficient permissions to view it

QlikView Publisher management console

Timely, accurate information delivery

- Full control over QlikView application and report distribution

- Automated data refresh

- Easy-to-create routines to guarantee quality, timeliness, and accuracy

Update QlikView documents with Publisher

"How do we keep our QlikView documents updated?"
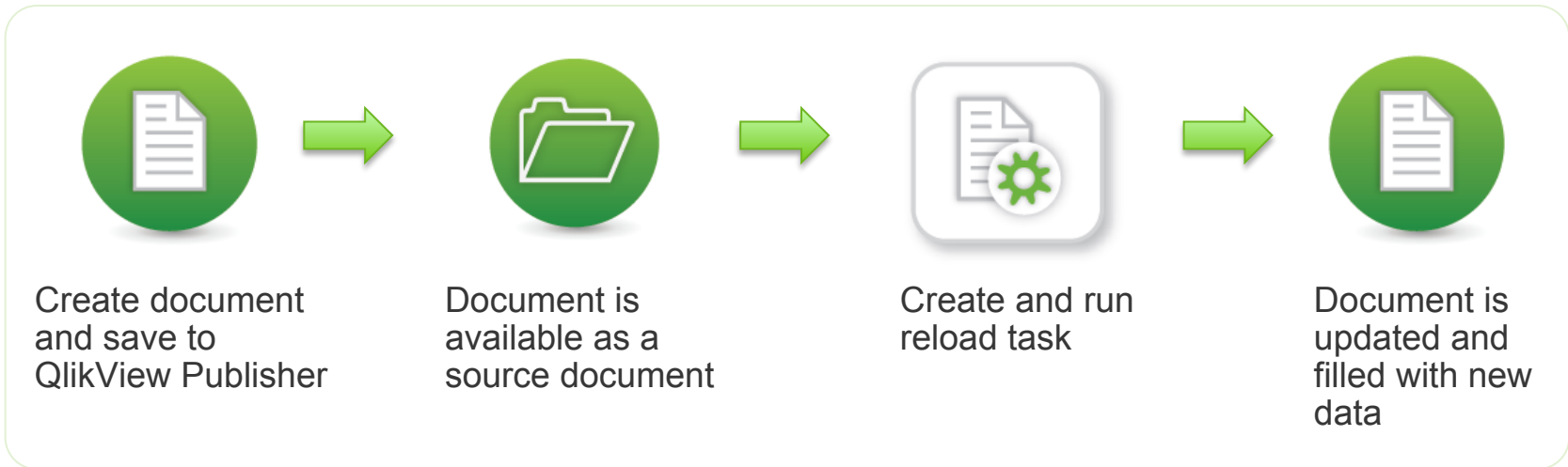
Create document and save to QlikView Publisher → Document is available as a source document → Create and run reload task → Document is updated and filled with new data

Benefits

- Flexible real-time administration provides short turnover times for adjustments
- Automated updating and personalization of information delivers the right information to the right recipient at the right time

Deliver personalized QlikView documents

"How do we deliver personalized QlikView documents throughout our organization?"

| Create document, save to QlikView Publisher. | Document is available as a source document. | Distribution task is created and personalization rules are defined. Schedule and run task. | User logs on to QlikView AccessPoint. Only documents they have access to are displayed. | Users analyze documents via QlikView Server. |
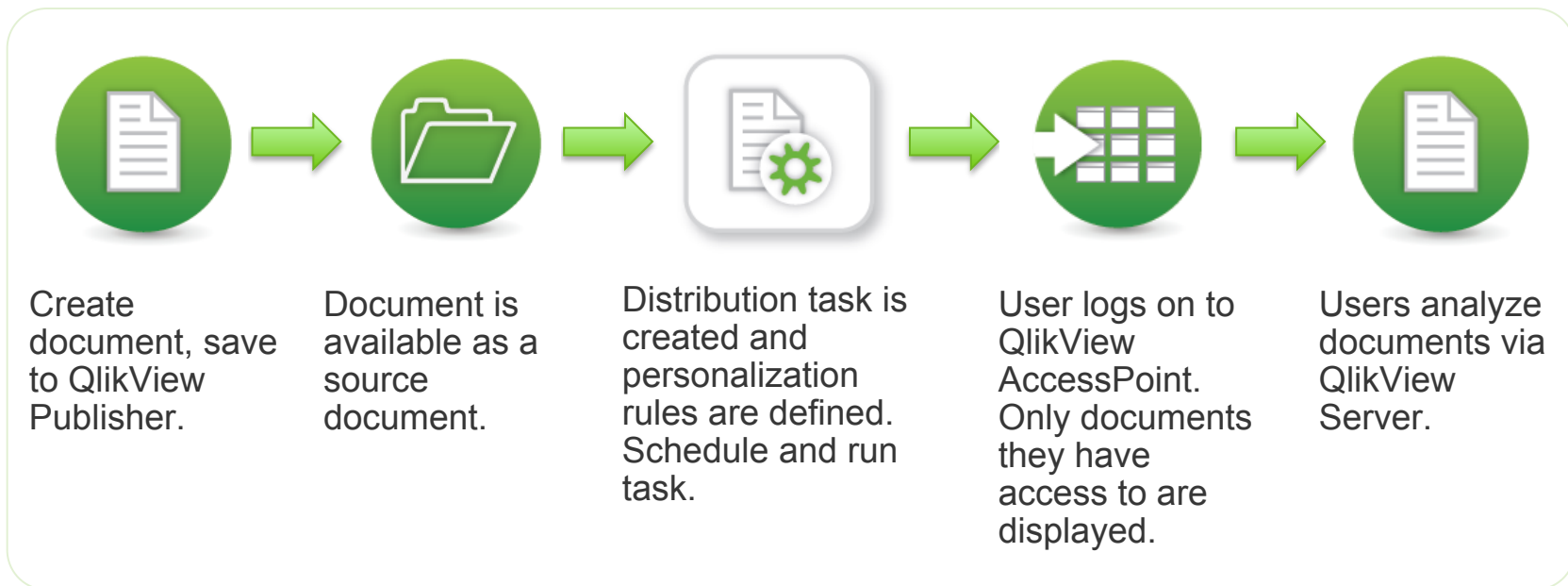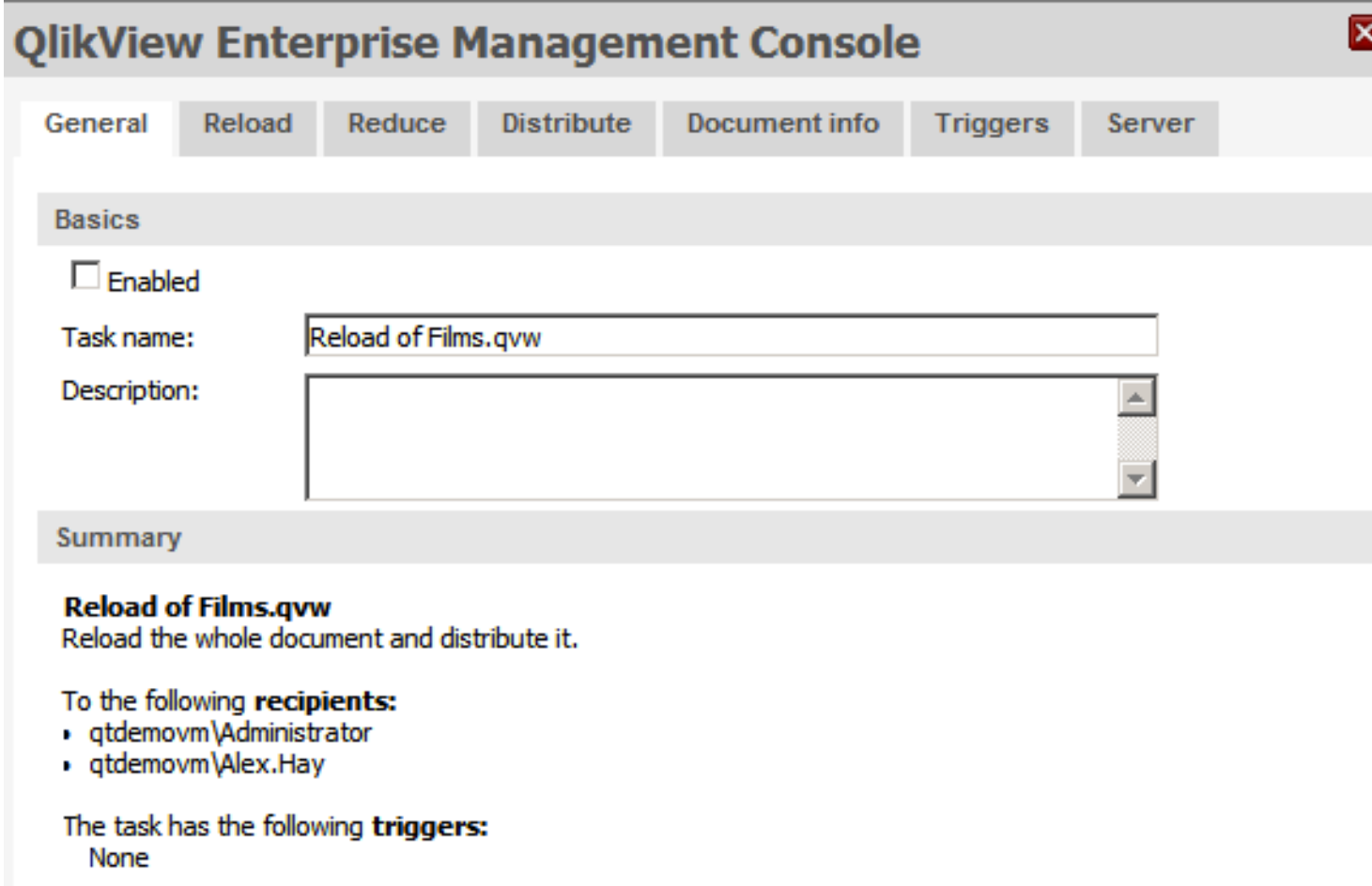
Benefits
- Each user is provided with his/her own information
- Personalized QlikView's are protected by network security

Easy-to-create routines guarantee quality and accuracy

QlikView Enterprise Management Console

General	Reload	Reduce	Distribute	Document info	Triggers	Server

Basics

☐ Enabled

Task name: `Reload of Films.qvw`

Description:

Summary

Reload of Films.qvw
Reload the whole document and distribute it.

To the following **recipients:**
- qtdemovm\Administrator
- qtdemovm\Alex.Hay

The task has the following **triggers:**
   None

Secure access to business data and analytics

- Slice QlikView applications into multiple QlikView documents
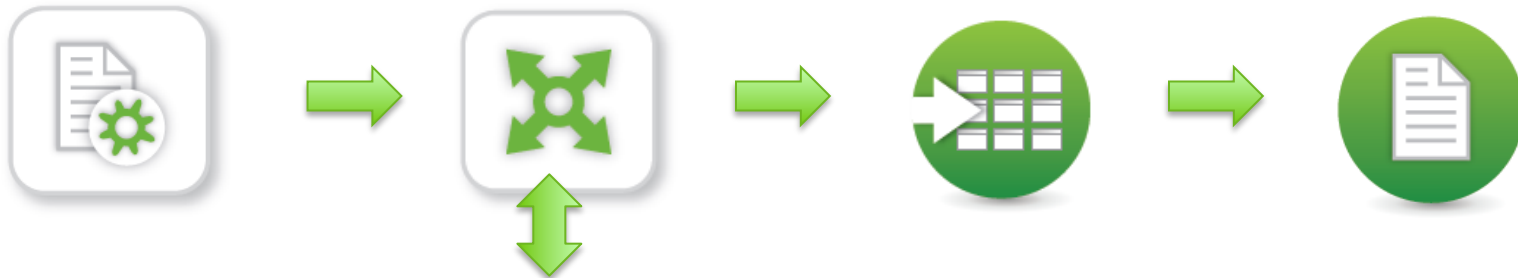
- Add row level security

- Distribute finalized QlikView documents or PDF reports to authorized users

- Integrate with existing enterprise security infrastructure

How to secure QlikView documents

"How can we secure our QlikView documents and provide single signon?"

Active Directory, Windows file security, 3rd party, native QlikView security

| Create and schedule distribution task. | Distribution service runs task, creates personalized document, retrieves user and group info from directory server. | User and group security is applied. Personalized document is published to AccessPoint. | User logs into AccessPoint. Security is enforced. User selects document from list and views it online via QlikView Server. |

Benefits

- Strong security leveraging existing security infrastructure
- Single point of control
- Single signon simplifies the experience for end users

Delegated administration

- Admin can delegate rights to create and manage tasks

- Full auditing

- Document administrators:

  - Work with limited subset of the management interface

  - Can be limited to working with a subset of documents

Enterprise-wide PDF report distribution

- Reload data, generate multiple PDFs, deliver reports to users' inboxes or specified network locations

- Wizard-based

- No user limits

- Optional add-on (additional license required)

How to deliver PDF QlikView reports

"How do we deliver personalized PDF reports to our non-QlikView users?"

| Create document and save to QlikView Publisher. | → | Document is available as a source document. | → | Distribution task is created to generate and email PDFs to users. Run the task. | → | User receives email containing PDF report. |

Benefits

- Flexible real-time administration provides short turnover times for adjustments
- Automated updating and personalization of information delivers the right information to the right recipient at the right time in the right format

PDF reports extend QlikView's reach

Weekly Report 04/30/2009

04/29/2009 Open 49.55 Closed 49.78 0.46% Volume 2,491,300

Volume
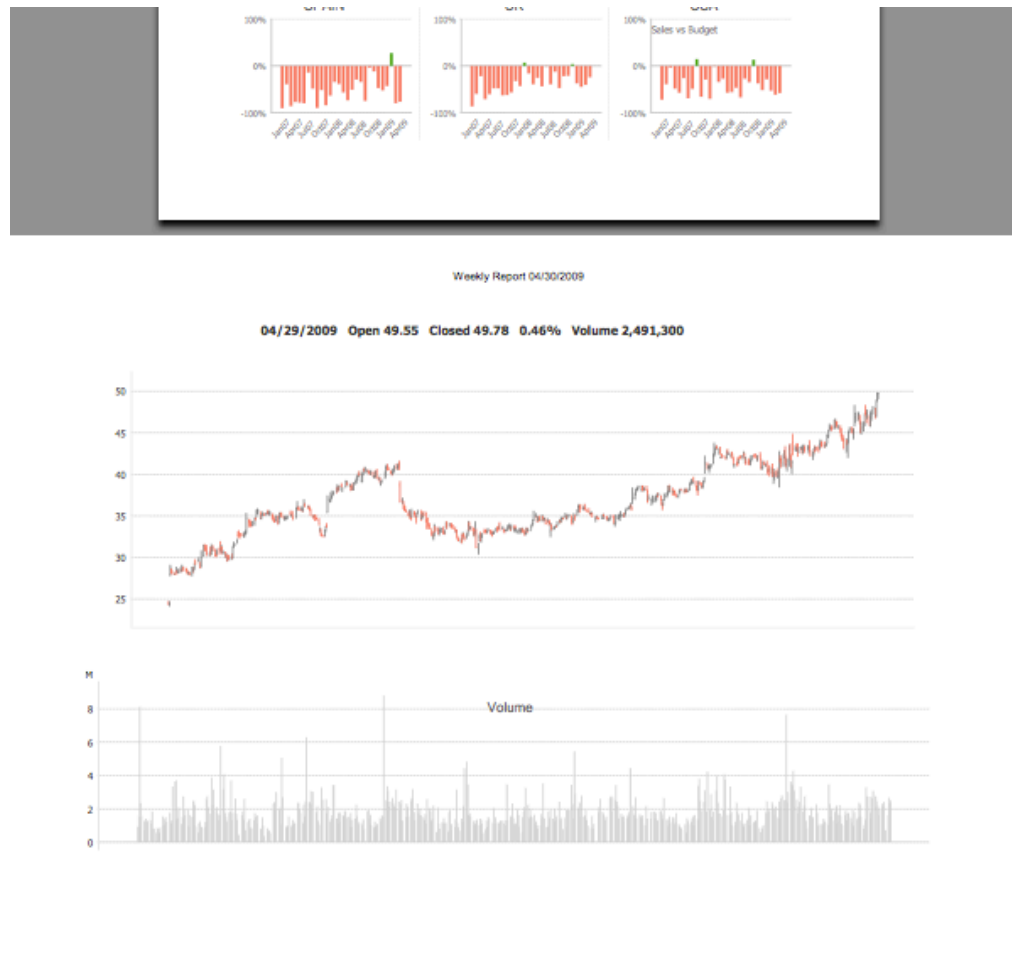
QlikView Publisher delivers scalability and manageability

- Service oriented architecture (Web services technology)

- Clustering for load-sharing and balancing

- Integration with enterprise system monitoring tools via SNMP

- Ability to trigger Publisher tasks using external enterprise scheduling tools

Control processes with QlikView Publisher

"How do we improve the visibility and control of our QlikView processes?"

Challenges
- Improve visibility of processes for more proactive system management
- Introduce one consolidated error reporting messenger to get one version of the truth at all times

Solution
- QlikView deployment with:
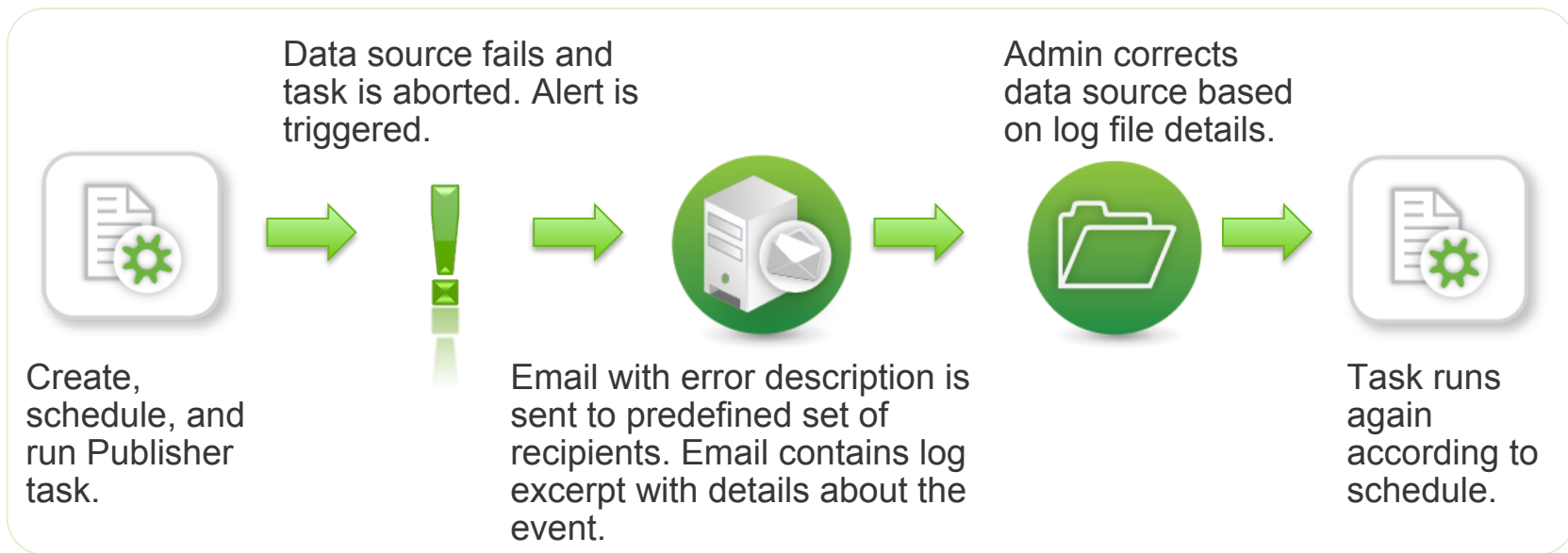 - Event-driven execution
  - Alert emails

Benefits
- Flexible real-time administration
- Event-driven execution starts processes directly from databases
- Alert emails notify the right people in time

Set up Publisher exception alerts

"How do set up QlikView Publisher to generate exception reports?"

Data source fails and task is aborted. Alert is triggered.

Admin corrects data source based on log file details.

Create, schedule, and run Publisher task.

Email with error description is sent to predefined set of recipients. Email contains log excerpt with details about the event.

Task runs again according to schedule.

Benefits

- Actionable information when an event occurs
- Targeted email alerts to the right people with the right information
- Proactive administration of QlikView documents

Enforce QlikView document dependencies

"How do we set up QlikView to handle document dependencies?"

Create and schedule reload task.

Distribution task is created and scheduled. Reload task must complete successfully before distribution task runs.

Reload task runs and fails.

Dependencies are enforced and the distribution task will not run.

Benefits

- Create natural relationships between documents
- Control the flow of information
- Ensure that data in QlikView documents is accurate

Technical specifications for QlikView Publisher

	32-bit (x86)	64-bit (x64)
Platform	Windows XP Professional[1] Windows Vista[1] Windows 7[1] Windows Server 2003 Windows Server 2008	Windows XP Professional x64 Edition[1] Windows Vista x64[1] Windows 7[1] Windows Server 2003 x64 Edition Windows Server 2008 x64 Edition Windows Server 2008 R2
Processor	Intel Core Duo compatible or higher recommended	Multi-core x64 compatible processors
Memory	1 GB minimum. Depending on data volumes more may be required.[2]	4 GB minimum. Depending on data volumes more may be required.[2]
Repository database	Native XML SQL Server 2005 or 2008	Native XML SQL Server 2005 or 2008

QlikView Publisher

- The right information

- To the right user

- At the right time

- In the right format