Consulting Services

JobScheduler
Architecture Decision Template

Information for Consulting Parties

Contents

Consulting Services

- **Decision Making**
  - Architecture Decisions

- **Supported Platforms**
  - Platforms: JobScheduler Master / JobScheduler Agent

- **Passive Cluster**
  - Architecture 1: Primary JobScheduler
  - Architecture 1: Backup JobScheduler

- **Active Cluster**
  - Architecture 2: Active Cluster JobScheduler
  - Architecture 2: Active Cluster JobScheduler with failed instance

- **Master / Agent Cluster**
  - Architecture 3: Master/Agent Passive Cluster JobScheduler
  - Architecture 3: Master/Agent Active Cluster JobScheduler

- **Supervisor JobScheduler**
  - Architecture 4: Supervisor for Passive Cluster
  - Architecture 4: Supervisor for Active Cluster
  - Architecture 4: Supervisor for Master/Agent Cluster
  - Architecture 4: Supervisor for Unclustered JobScheduler

Architecture Decisions

Passive Cluster
- Primary & Backup JobScheduler
- Redundancy and automated fail-over

Active Cluster
- Active Cluster JobScheduler
- Redundancy and load sharing

Master/ Agent Cluster
- Master/Agent Cluster JobScheduler
- Redundancy, load sharing, load distribution
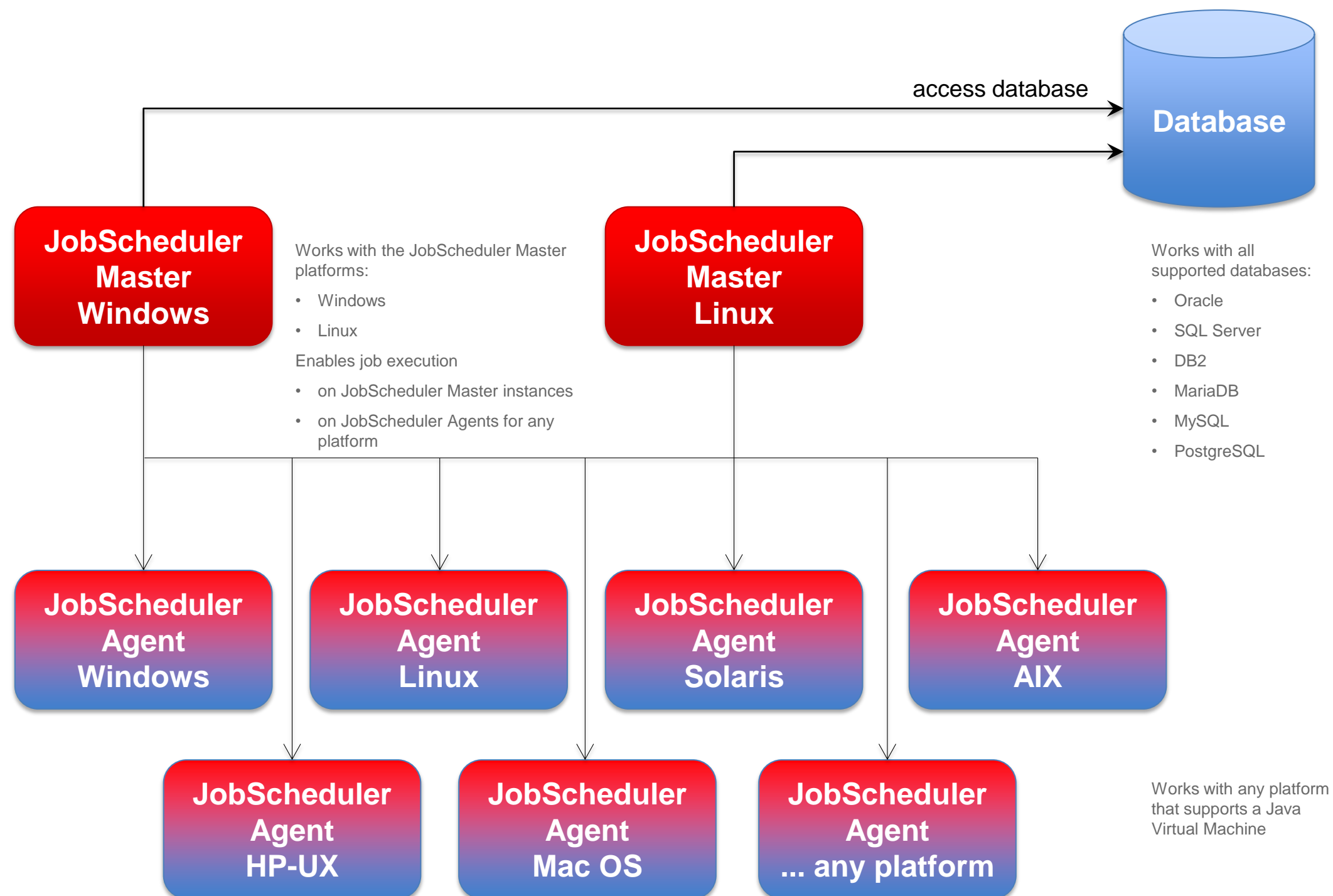
Supervisor JobScheduler
- Passive & Active Cluster Support,
- Master/Agent Cluster Support, Unclustered JobScheduler Support
- Central Configuration

Architecture Decision Templates: Supported Platforms

Master/Agent Platforms

- JobScheduler Master uses a database that is located on any platform
- JobScheduler Master is available for Windows and Linux
- JobScheduler Agents are available for any platform that supports a Java Virtual Machine

Job Execution

- Jobs are executed locally on the JobScheduler Master.
- Jobs are executed on a remote JobScheduler Master instance
- Jobs are executed on any JobScheduler Agent.
- JobScheduler Agents enable remote file watching, i.e. they trigger job starts in the JobScheduler Master for incoming files

JobScheduler Master Windows

Works with the JobScheduler Master platforms:
- Windows
- Linux

Enables job execution
- on JobScheduler Master instances
- on JobScheduler Agents for any platform

JobScheduler Master Linux

access database

Database

Works with all supported databases:
- Oracle
- SQL Server
- DB2
- MariaDB
- MySQL
- PostgreSQL

JobScheduler Agent Windows

JobScheduler Agent Linux

JobScheduler Agent Solaris

JobScheduler Agent AIX

JobScheduler Agent HP-UX

JobScheduler Agent Mac OS

JobScheduler Agent ... any platform

Works with any platform that supports a Java Virtual Machine

Architecture 1: Primary JobScheduler

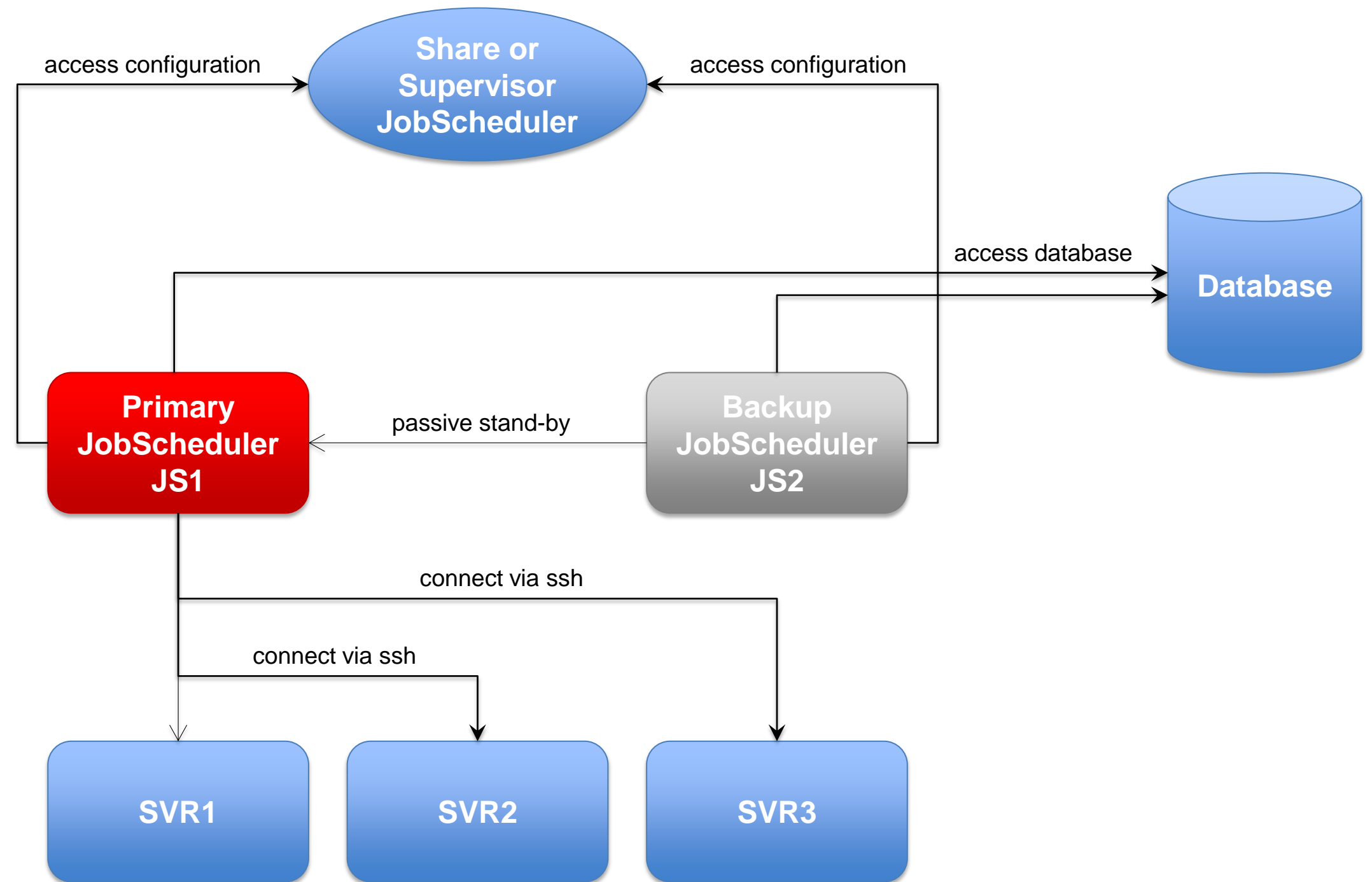Architecture Decision Templates: Passive Cluster

Passive Cluster

- Primary and Backup JobScheduler use the same database
- Primary JobScheduler is monitored by its failover instance
- Failover instance operates in stand-by mode
- All connections to servers use the SSH protocol

SSH Connections

JITL Jobs

- Requires a JVM per task
- Memory resources

SSH Client

- No pre-/post-processing
- No substitution of parameters in script files
- Script files have to be provided on the target system

access configuration → **Share or Supervisor JobScheduler** ← access configuration

access database → **Database**

Primary JobScheduler JS1 ← passive stand-by — **Backup JobScheduler JS2**

connect via ssh

connect via ssh

SVR1   **SVR2**   **SVR3**

Architecture 1: Backup JobScheduler

Architecture Decision Templates: Passive Cluster
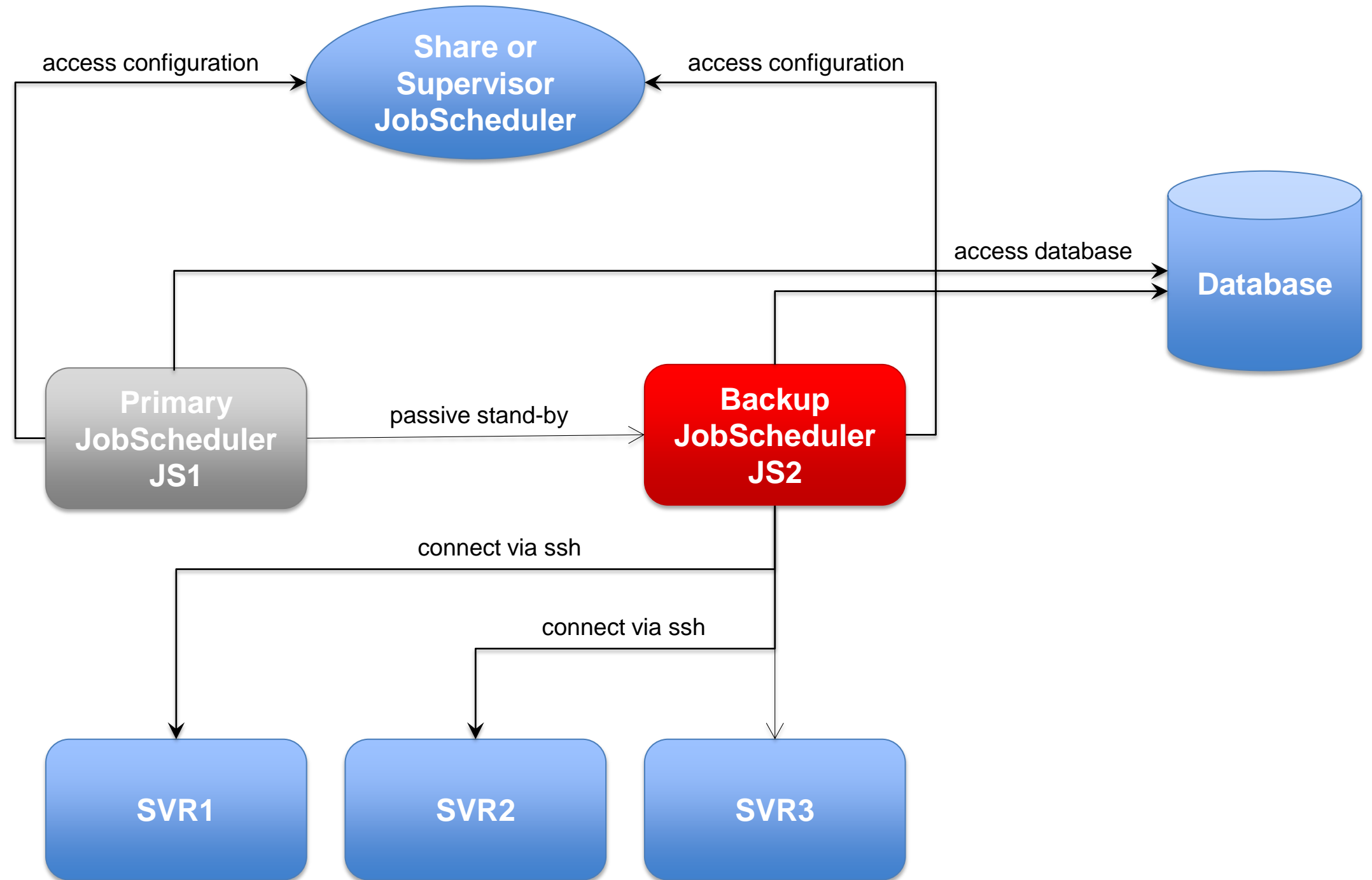
Passive Cluster
- Primary and Backup JobScheduler use the same database
- Backup JobScheduler is active after failure of Primary instance
- Primary instance operates in stand-by mode
- All connections to servers use the SSH protocol

SSH Connections

JITL Jobs
- Requires a JVM per task
- Memory resources

SSH Client
- No pre-/post-processing
- No substitution of parameters in script files
- Script files have to be provided on the target system

Architecture 2: Active Cluster JobScheduler

Architecture Decision Templates: Active Cluster
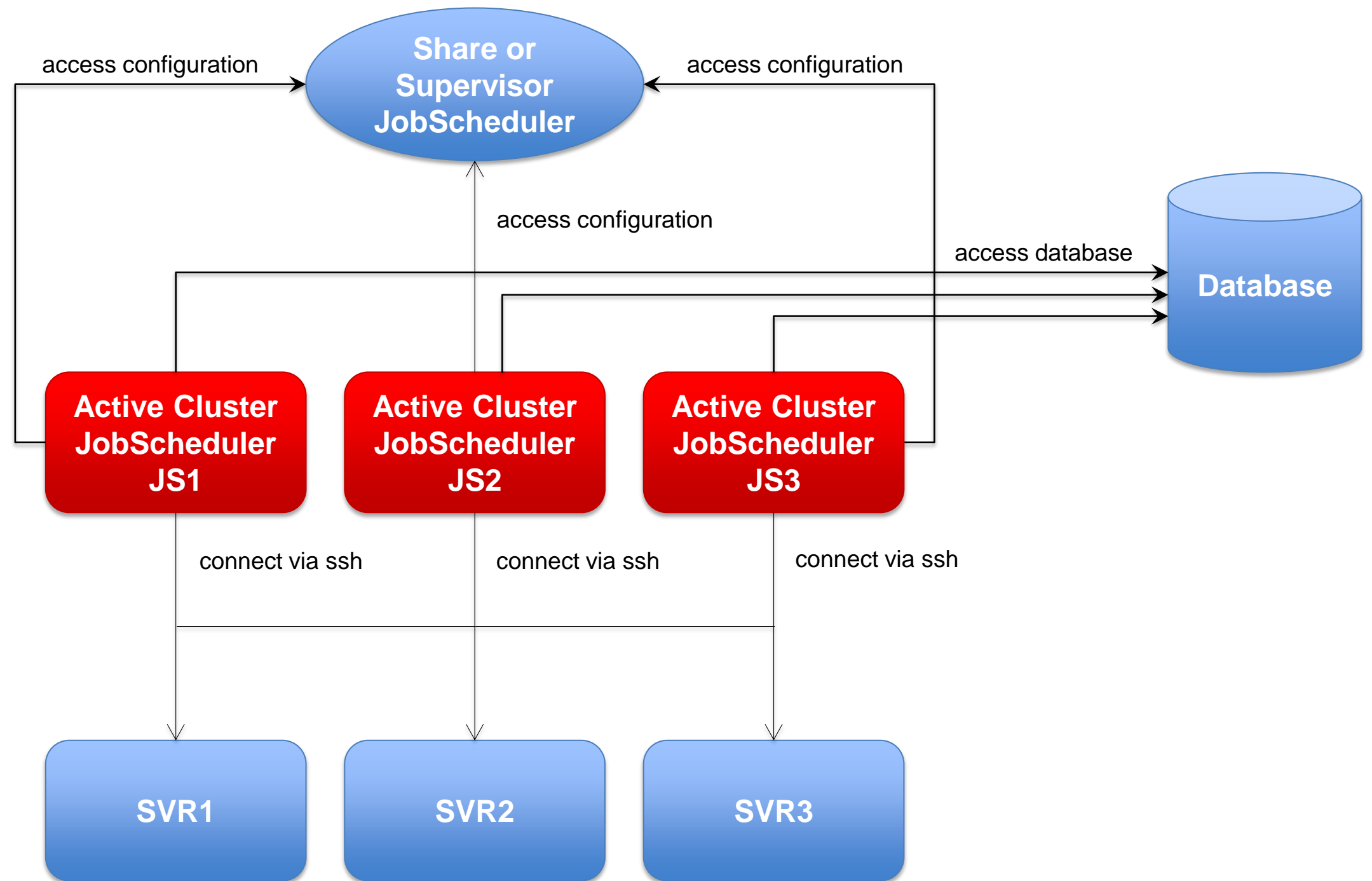
Active Cluster
- Cluster JobSchedulers use the same database
- Cluster JobSchedulers share the workload of jobs
- All Instances operate in active mode
- All connections to servers use the ssh protocol

SSH Connections
JITL Jobs
- Requires a JVM per task
- Memory resouces
SSH Client
- No pre-/post-processing
- No substitution of parameters in script files
- Script files have to be provided on the target system

Architecture 2: Active Cluster JobScheduler with failed instance

Architecture Decision Templates: Active Cluster
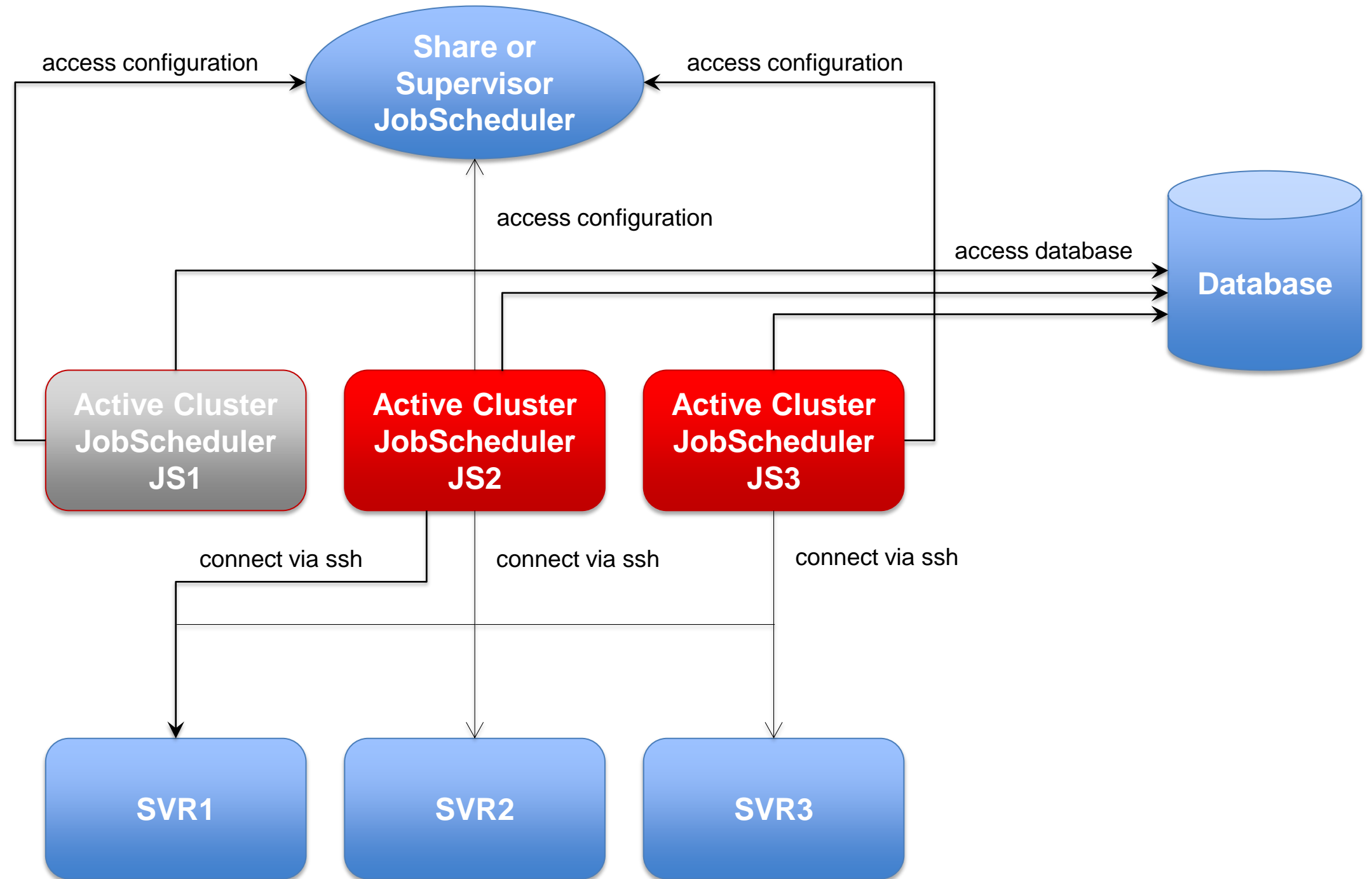
Active Cluster
- Cluster JobSchedulers use the same database
- Cluster JobSchedulers share the workload of jobs
- All Instances operate in active mode
- All connections to servers use the ssh protocol

SSH Connections
JITL Jobs
- Requires a JVM per task
- Memory resources
SSH Client
- No pre-/post-processing
- No substitution of parameters in script files
- Script files have to be provided on the target system

Share or Supervisor JobScheduler

access configuration

access configuration

access configuration

access database

Database

Active Cluster JobScheduler JS1

Active Cluster JobScheduler JS2

Active Cluster JobScheduler JS3

connect via ssh

connect via ssh

connect via ssh

SVR1

SVR2

SVR3

Architecture 3: Master/Agent Passive Cluster JobScheduler

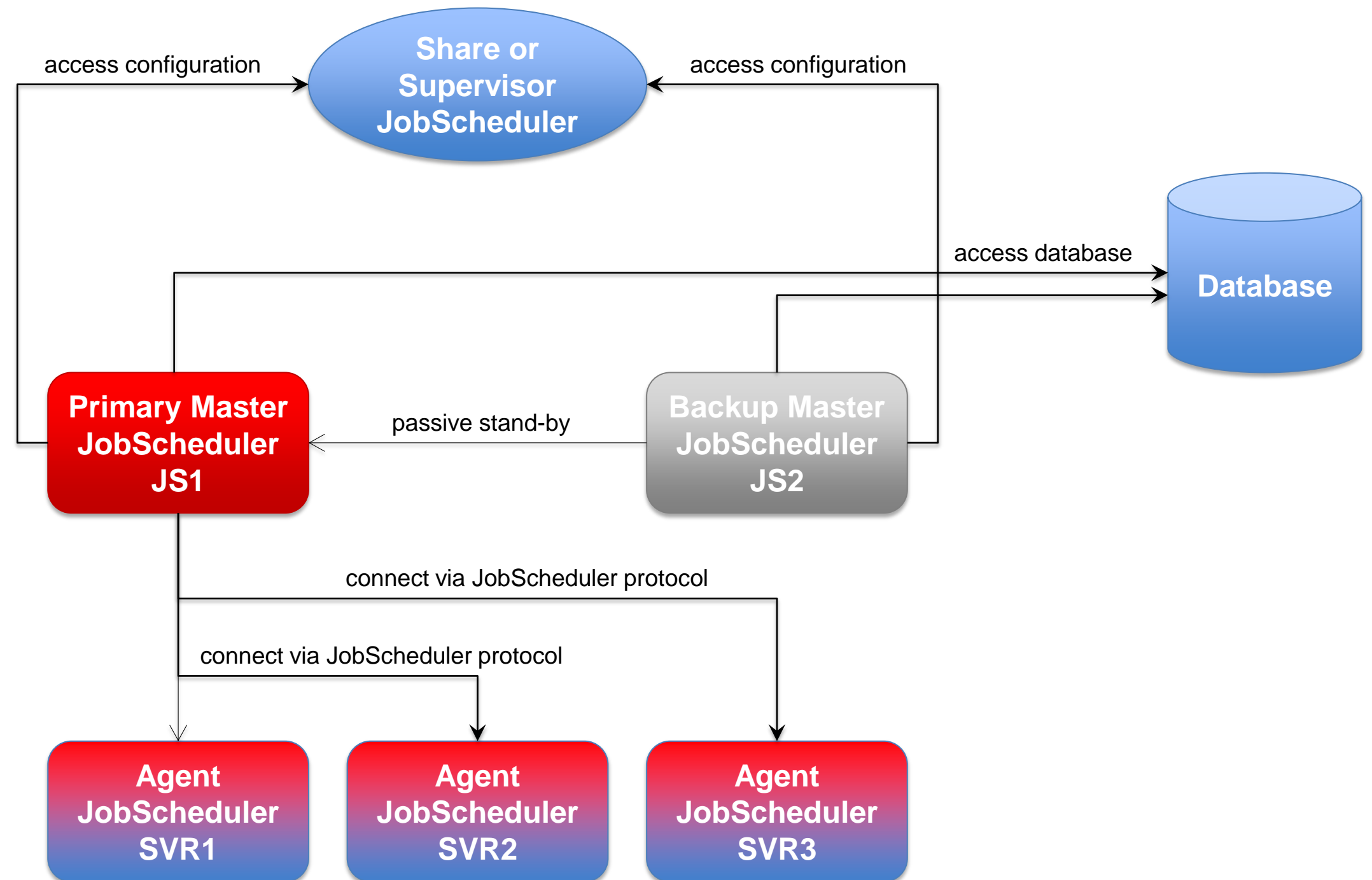Architecture Decision Templates: Master/Agent Passive Cluster

Master/Agent Passive Cluster

- Primary and Backup JobScheduler use the same database
- Primary JobScheduler is monitored by its Backup instance
- Backup instance operates in stand-by mode
- All Cluster instances use Agents to execute jobs on remote servers
- Connections to servers use the internal protocol

Job Execution

- Jobs are executed locally per JobScheduler Agent.
- No central resources required for job execution
- Pre-/post-processing
- Use of JITL Jobs or script files with parameter substitution

Share or Supervisor JobScheduler

access configuration

access configuration

access database

Database

Primary Master JobScheduler JS1

passive stand-by

Backup Master JobScheduler JS2

connect via JobScheduler protocol

connect via JobScheduler protocol

Agent JobScheduler SVR1

Agent JobScheduler SVR2

Agent JobScheduler SVR3

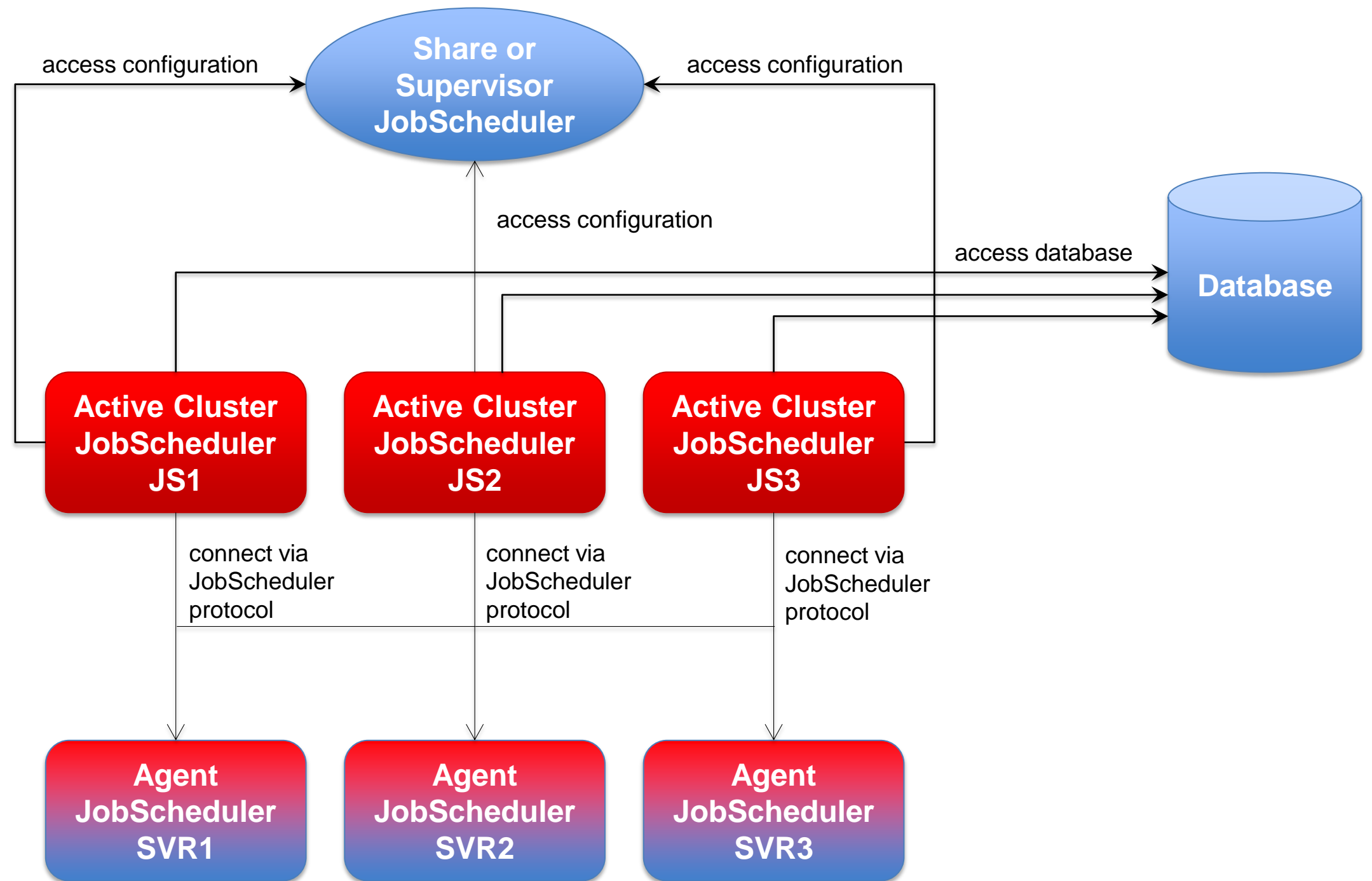Architecture Decision Templates: Master/Agent Active Cluster

Master/Agent Active Cluster
- Cluster JobSchedulers use the same database
- Cluster JobSchedulers share the workload of jobs
- All Instances operate in active mode
- All Cluster instances use Agents to execute jobs on remote servers

Job Execution
- Jobs are executed locally per JobScheduler Agent.
- No central resources required for job execution
- Pre-/post-processing
- Use of JITL Jobs or script files with parameter substitution

access configuration   access configuration

Share or Supervisor JobScheduler

access configuration

access database

Database

Active Cluster JobScheduler JS1

Active Cluster JobScheduler JS2

Active Cluster JobScheduler JS3

connect via JobScheduler protocol

connect via JobScheduler protocol

connect via JobScheduler protocol

Agent JobScheduler SVR1

Agent JobScheduler SVR2

Agent JobScheduler SVR3

Architecture 4: Supervisor for Passive Cluster

Architecture Decision Templates: Supervisor JobScheduler

Passive Cluster
- Primary and Backup JobScheduler use the same database
- Primary JobScheduler is monitored by its Backup instance
- Backup instance operates in stand-by mode
- All connections to servers use the ssh protocol

Supervisor JobScheduler
- Distribute configuration to Primary and Backup JobScheduler instances

Architecture 4: Supervisor for Active Cluster

Architecture Decision Templates: Supervisor JobScheduler
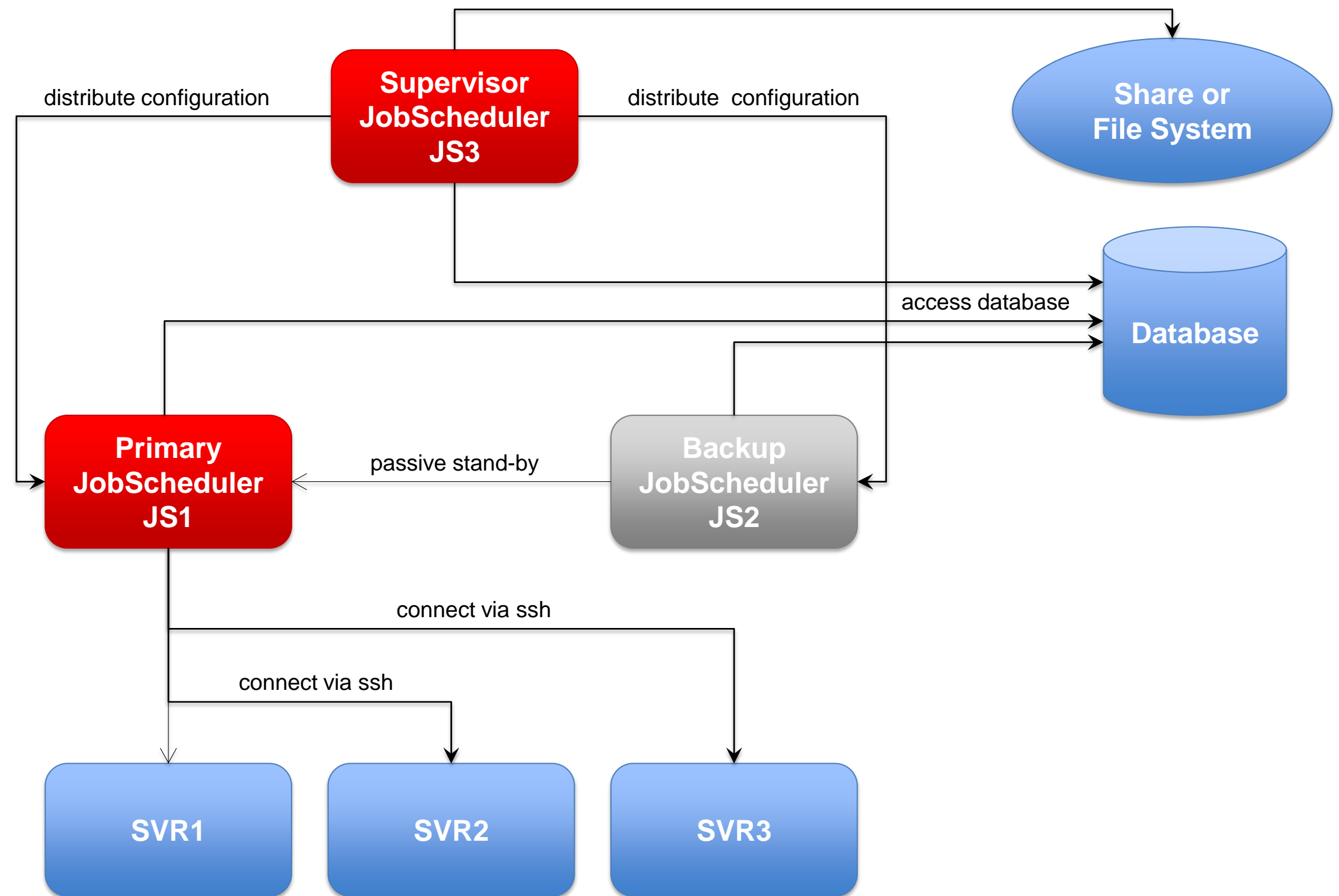
Active Workload JobScheduler Cluster
- Cluster JobSchedulers use the same database
- Cluster JobSchedulers share the workload of jobs
- All Instances operate in active mode
- All connections to servers use the ssh protocol

Supervisor JobScheduler
- Distribute configuration to Cluster JobScheduler instances

Architecture Decision Templates: Supervisor JobScheduler
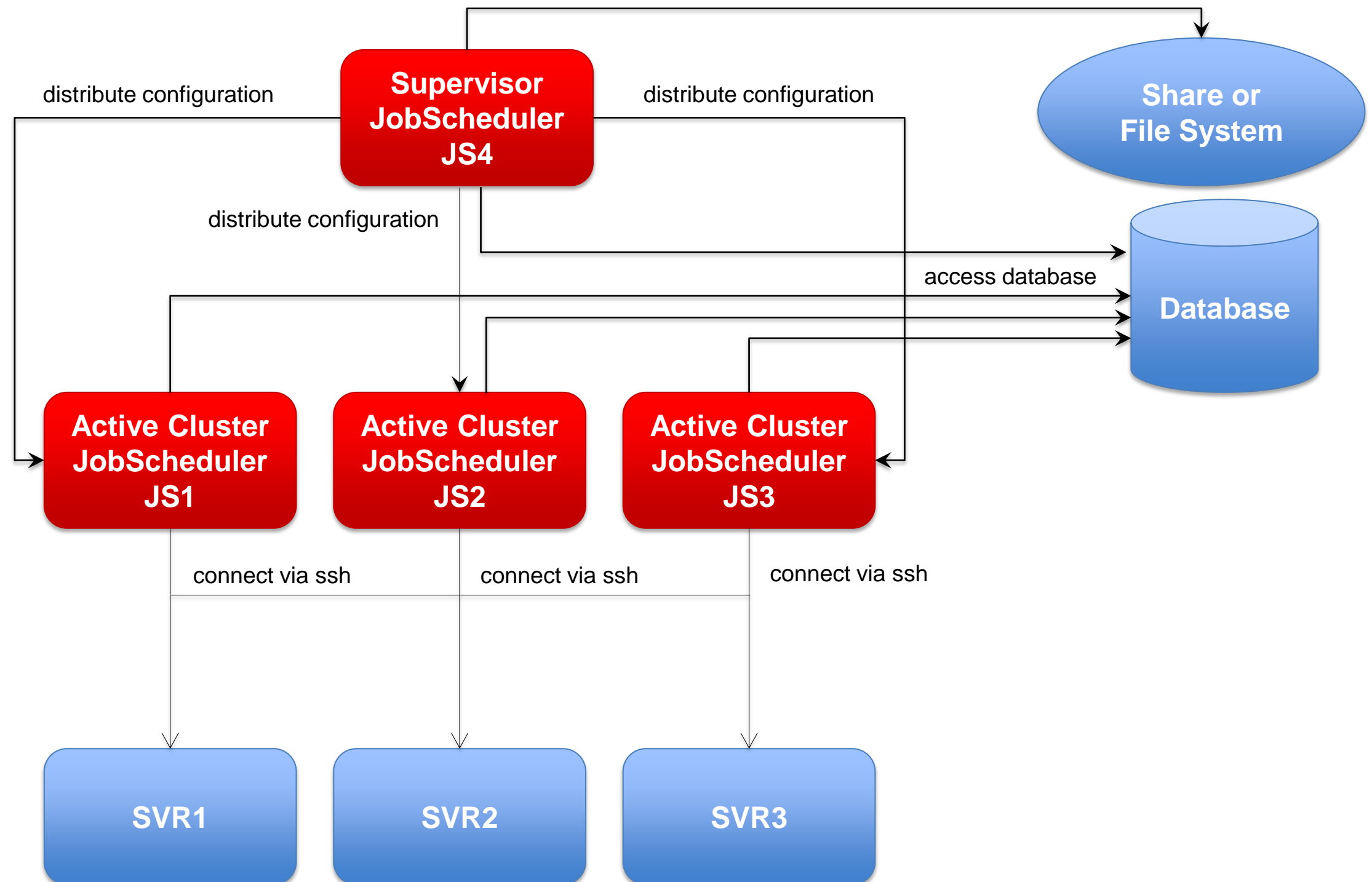
Master/Agent Active Cluster

- Cluster JobSchedulers use the same database
- Cluster JobSchedulers share the workload of jobs
- All Instances operate in active mode
- All Cluster instances use Agents to execute jobs on remote servers

Supervisor JobScheduler

- Distribute configuration to Cluster JobScheduler instances

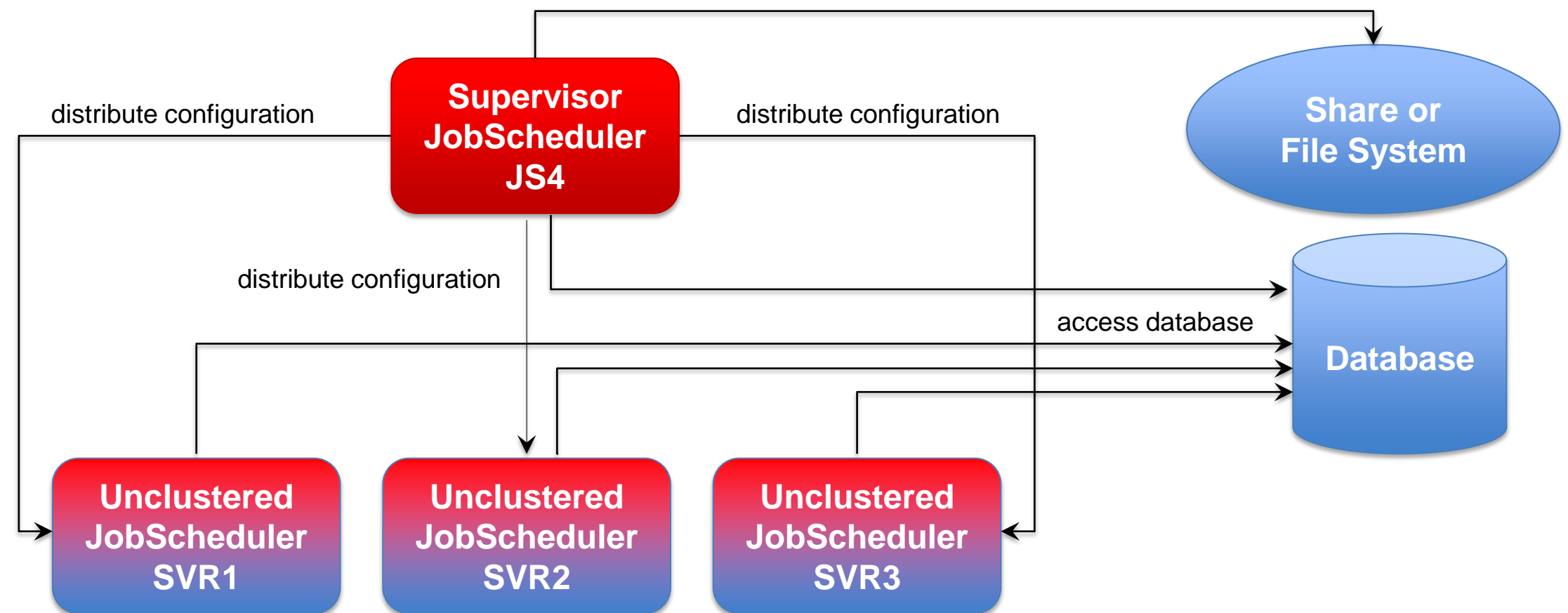Architecture Decision Templates: Supervisor JobScheduler
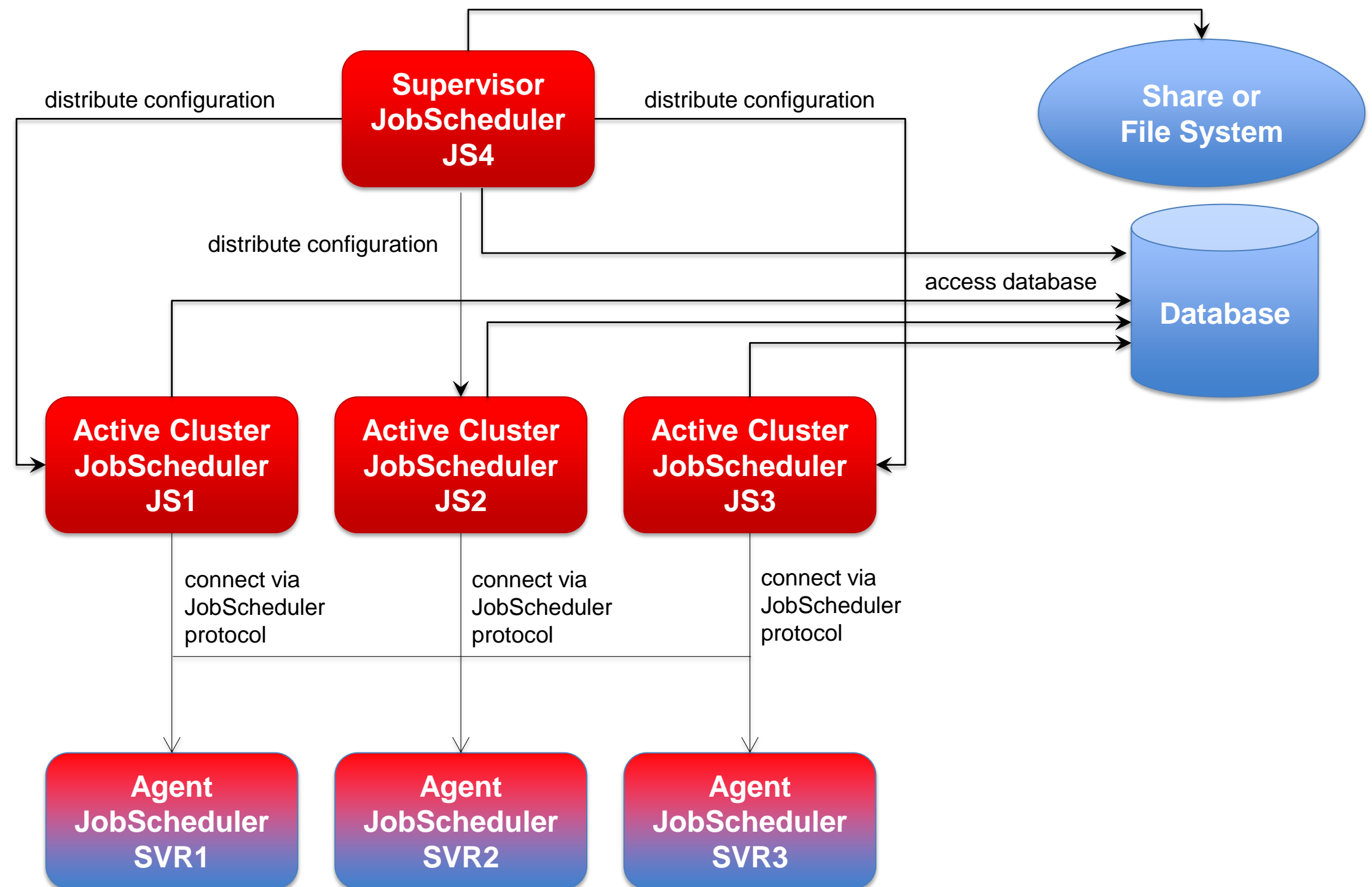
Unclustered JobSchedulers

- JobSchedulers use the same database
- JobSchedulers operate independently from each other
- All Instances operate in active mode

Supervisor JobScheduler

- Distribute configuration to JobScheduler instances

Software- und Organisations-Service

Questions?

Comments?

Feedback?

Software- und
Organisations-
Service GmbH

Giesebrechtstr. 15
D-10629 Berlin

info@sos-berlin.com
http://www.sos-berlin.com