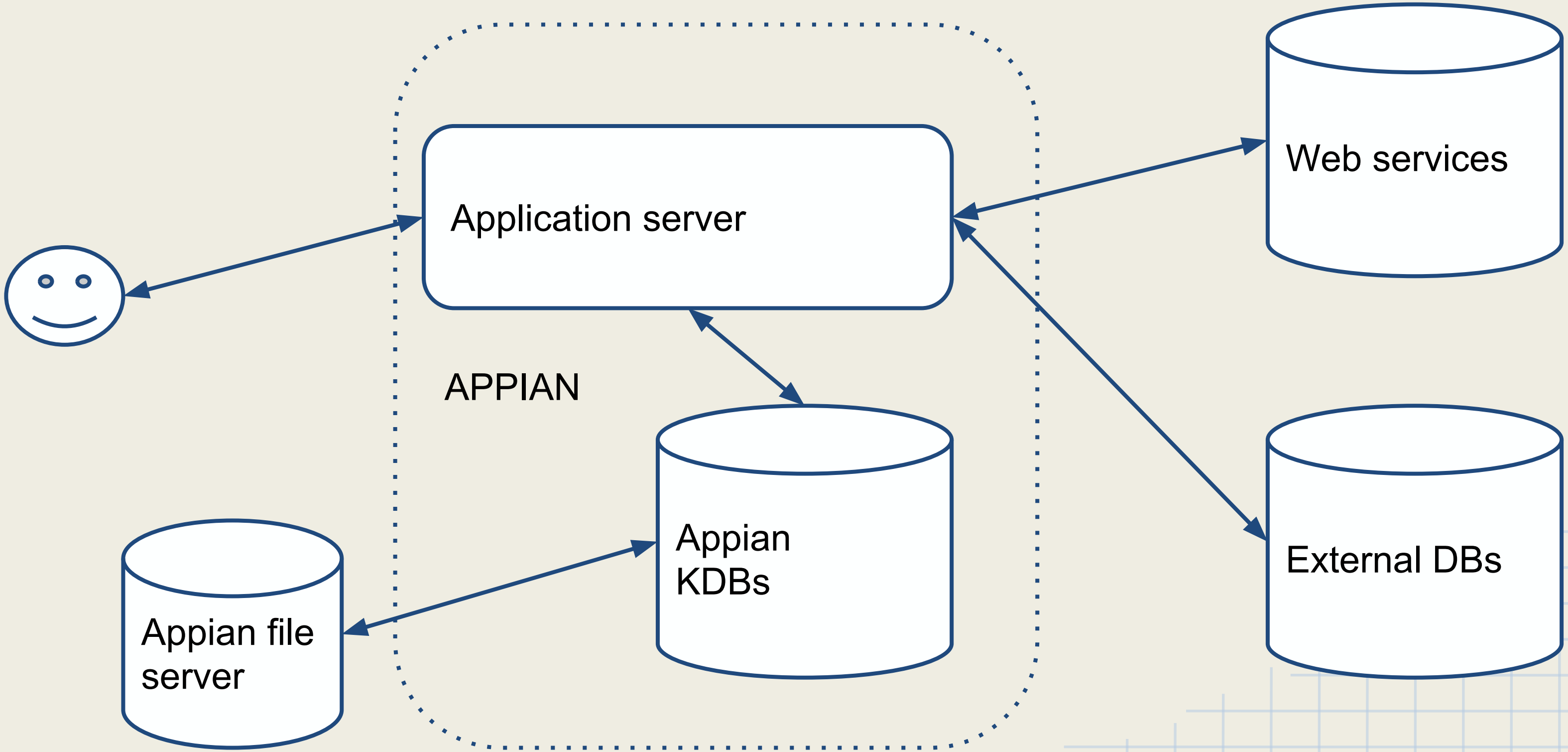
Appian Administration

Serafeim Papastefanos
213 137 1059 / 694 51 95 545
spapas@gmail.com

Architecture



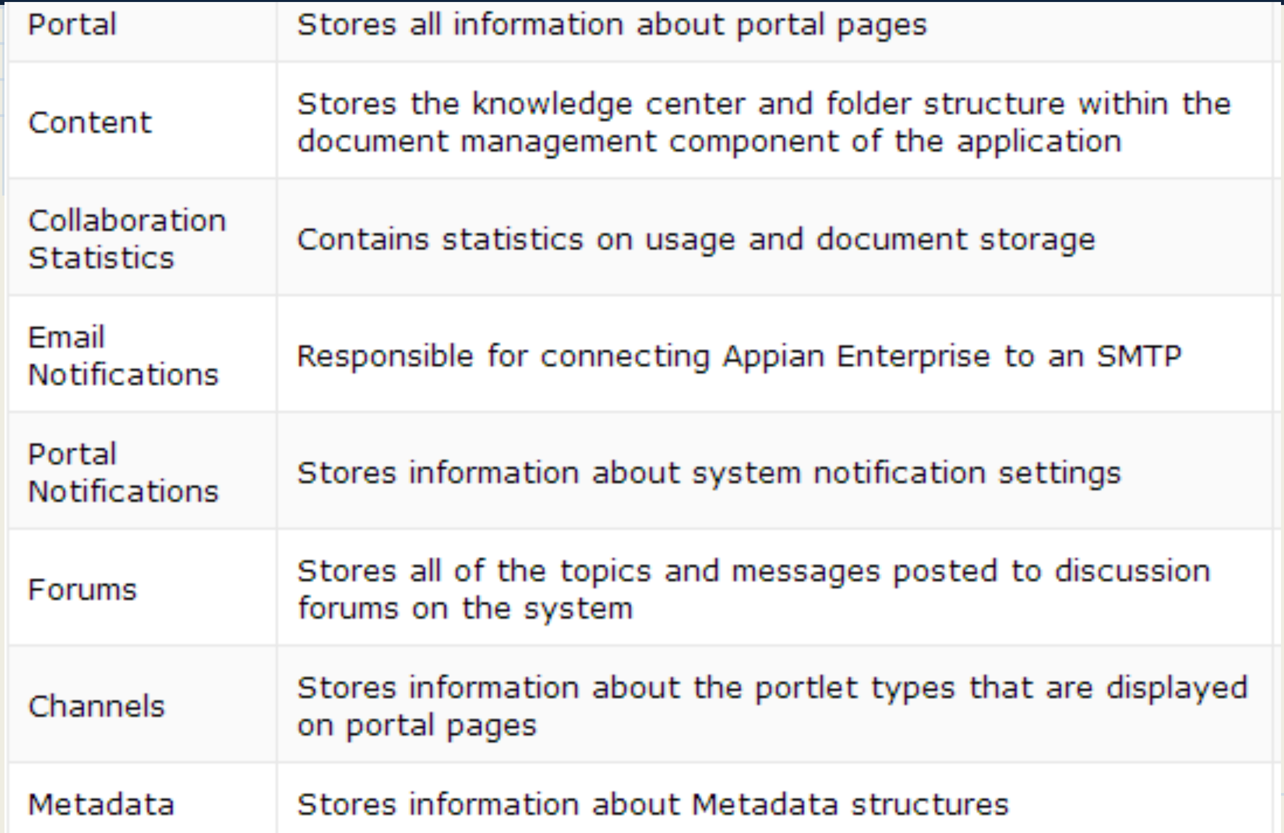
Appian Engines

- Real-Time-In Memory Databases
- Two processes: Gateway + database
  - API talks to gateway which serializes requests to database
- KDB files
  - Database
  - Synchronous Transaction Logs
- Loaded in RAM when engines start
- When a write occurs it is applied to the memory database and written to the transaction logs
- Logs are applied to the kdb file-database when a checkpoint occurs
- A checkpoint occurs when Appian is properly shut down
- If there is an improper shutdown, transaction logs will be applied when Appian is started - so it will need some time

Checkpointing

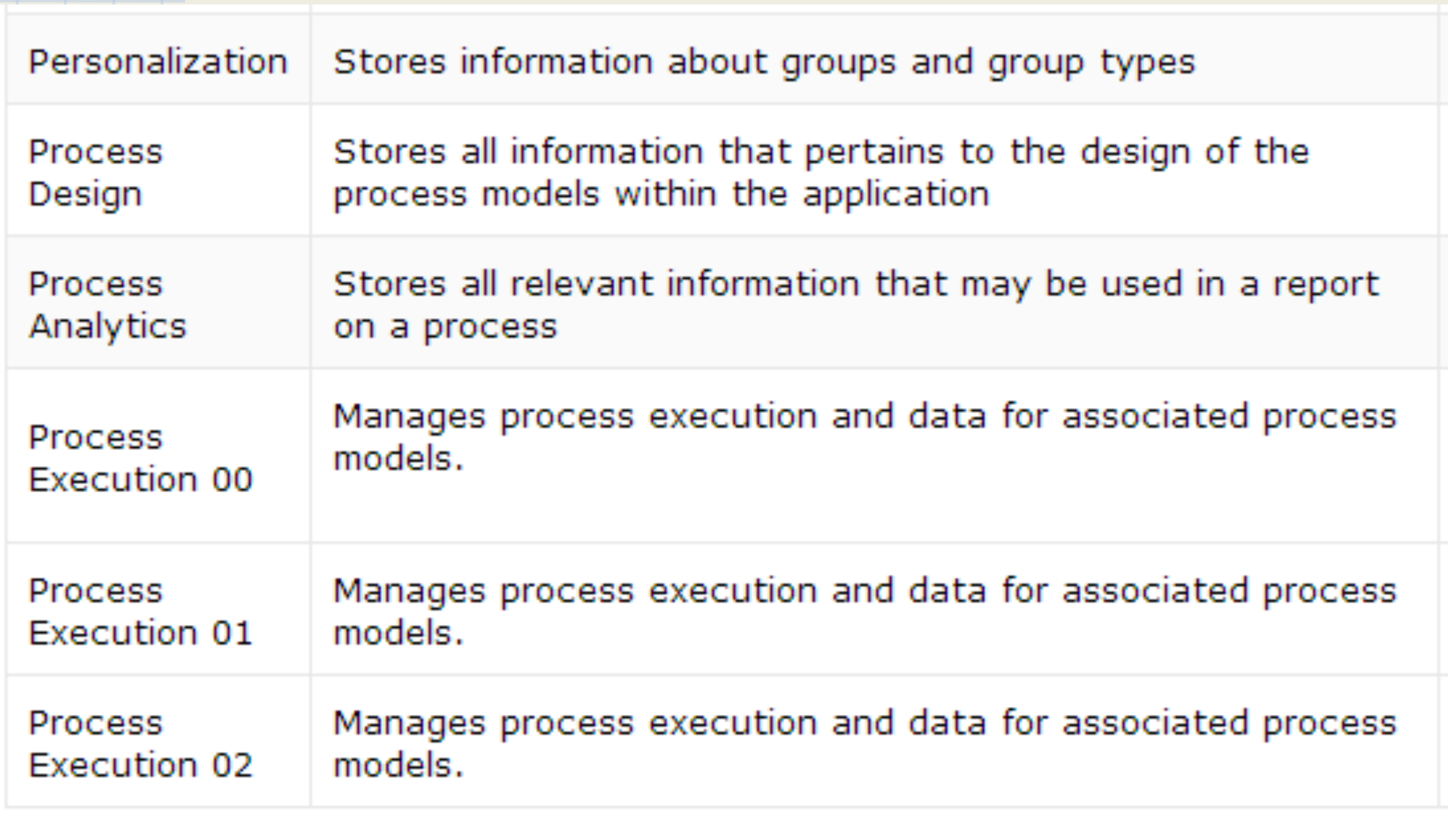
- Each KDB file has a name of xx_yyy.kdb, where xx is the name of the engine (PX - process execution, PD - process design etc) and yyy is the current database number.
- A new kdb file with an increased by one number is created when the checkpoint finishes - size of new kdb will be decreased (because the log will be deleted)
- For instance if we have px_111.kdb, after the checkpoint we will have both px_112.kdb and px_111.kdb
- The kdb with the biggest number is our current database
- When the engines are checkpointing, a writing_*.kdb file will be created - this will be renamed to the new kdb.
- If we have writing_*.kdb files then checkpointing didn’t finish successfully (not a real problem)
- Not all engines have the same numbers
- Engines are not responsive during the checkpointing

Analysis of Appian Engines 1



Portal	Stores all information about portal pages
Content	Stores the knowledge center and folder structure within the document management component of the application
Collaboration Statistics	Contains statistics on usage and document storage
Email Notifications	Responsible for connecting Appian Enterprise to an SMTP
Portal Notifications	Stores information about system notification settings
Forums	Stores all of the topics and messages posted to discussion forums on the system
Channels	Stores information about the portlet types that are displayed on portal pages
Metadata	Stores information about Metadata structures

Analysis of Appian Engines 2



Personalization	Stores information about groups and group types
Process Design	Stores all information that pertains to the design of the process models within the application
Process Analytics	Stores all relevant information that may be used in a report on a process
Process Execution 00	Manages process execution and data for associated process models.
Process Execution 01	Manages process execution and data for associated process models.
Process Execution 02	Manages process execution and data for associated process models.

Distributed architectures

- Different servers for appian engines & application server
  - Configuration with topology.xml
  - Our current architecture
- Multiple Application servers communicating with the same appian engines
  - Easy to do it (only application servers need configuration)
  - A load balancer is needed in front
  - Minimum risk - we'd done it
- Appian engines in different servers (not replicated)
  - For instance, one server for the process execution engines, one server for the process analytics engines, a third for every other engine
  - Can also add more execution & analytics engines
  - We have not tried it (yet) - however risk is minimum and benefits are great - *if* the bottleneck is on appian engines
- Appian engines in different servers (replicated)
  - :-(

Process Execution & Analytics engines

- Three execution engines
- Three different counters for Process Ids
  - ~ 260M difference in numbers
- Three analytics engines
- Reports are generated sequentially on each engine
- There is a limit on the amount of generation time for each report (~10 s)
- If one user requests a page with three long running reports (~10 s) then a second user request in the same page will start processing after 10 s
  - So it’d need 20 s to complete !

Check status of appian engines

- <appian>\server_scripts\diagnostic\checkengine.bat
  - S U M M A R Y (2013-11-01 11:10:32.708 GMT) S U M M A R Y
  - _APP_ _ACTIVE_GW_ _STATUS_ _ISSUES_
  - Discussion Forums 1/1 Okay
  - Notifications Service 1/1 Okay
  - Notifications Email Processor 1/1 Okay
  - Channels 1/1 Okay
  - Collaboration 1/1 Okay
  - Collaboration Statistics 1/1 Okay
  - Personalization 1/1 Okay
  - Portal 1/1 Okay
  - Metadata Structure Service 1/1 Okay
  - Process-Design 1/1 Okay
  - Process-Analytics0000 1/1 Okay
  - Process-Analytics0001 1/1 Okay
  - Process-Analytics0002 1/1 Okay
  - Process-Exec00 1/1 Okay
  - Process-Exec01 1/1 Okay
  - Process-Exec02 1/1 Okay
- All engines should return Okay

Start/stop appian engines

- Production:
    - Use windows services
- Uat / dev:
    - <appian>\server_scripts\start-suite.bat
    - <appian>\server_scripts\stop-suite.bat
- When starting:
    - Run diagnostic/checkengine.bat until all engines are ok. Only then start application server
- When stopping: First stop Application server
    - Then run stop-suite and diagnostic\checkengine.bat until you see the message * FATAL! No gateway is active. to *all* engines
    - Stop-suite may need to run a second or third time if kdbs are too big
- Try to not kill appian processes (k.exe) abruptly

Application server

- We are using WebSphere
  - Start and stop through windows services
- Two main applications
  - appian.suite: The appian web app - uses a Java API to talk to the engines
  - Root (/): This hosts a number of JSPs that are used to call web processes or to return JSON content for usage in Ajax.
- Various other components
  - Datasources
  - Messaging with JMS
  - Email poller
  - Spring-security

Logging

- Very detailed logs (https://forum.appian.com/suite/wiki/71/Logging)
- <appian>/logs
  - login-audit.csv: All logins to the system
  - gw_XX_date.log: Logs for gateway XX(PD, PX, etc)
  - db_XX_date.log: Logs for database XX
  - deletion.log: Who and when deletes objects from designer
  - mail-listener.log: Log of email poller
- <appian>/logs/data-metrics
- <appian>/logs/perflogs
- Most errors and log output are visible only to the application server logs since it is executed through a java thread there
  - Appian engines don’t actually use java at all !
  - http://en.wikipedia.org/wiki/K_(programming_language)