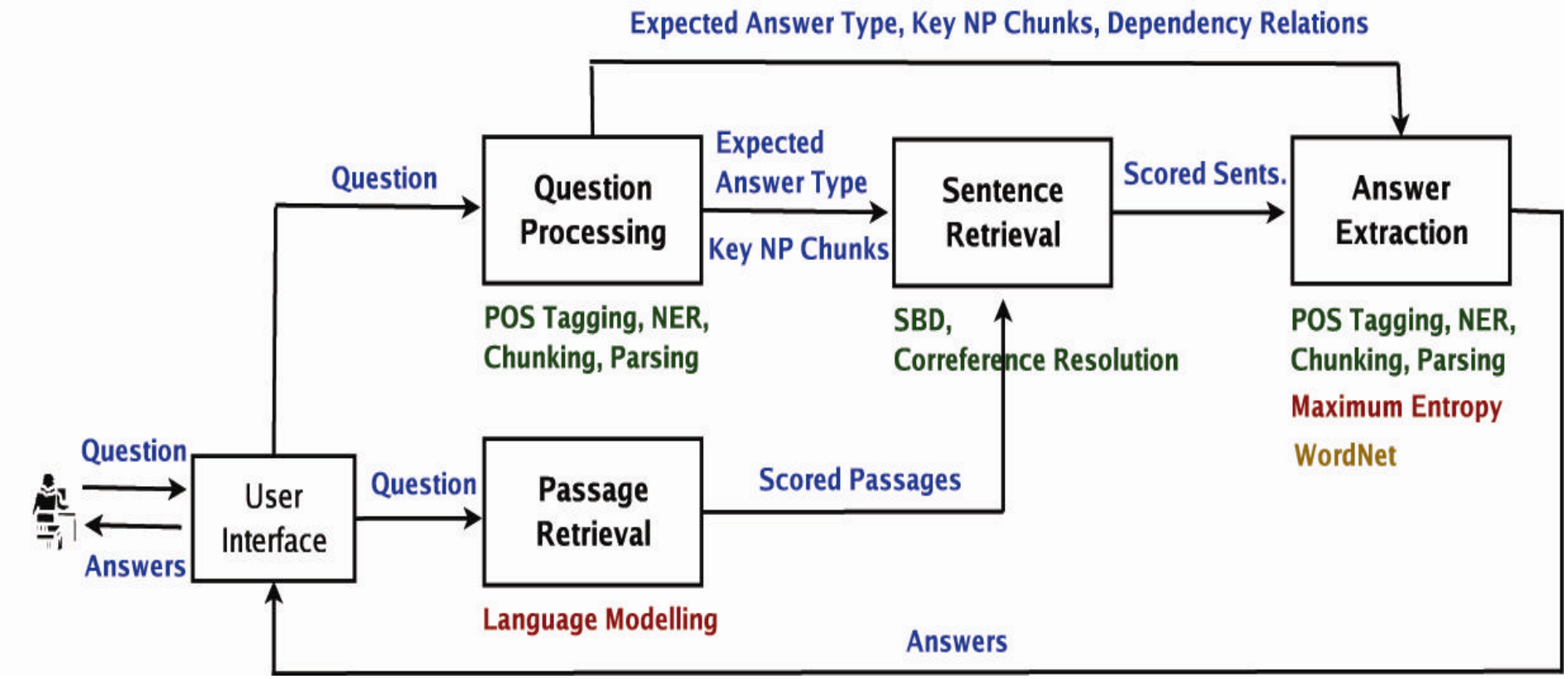
A Framework-based Online Question Answering System

Oliver Scheuer, Dan Shen, Dietrich Klakow

Outline

- General Structure for Online QA System
- Problems in General Structure
- Framework-based Online QA system
  - Requirements of Online QA system
  - Framework Architecture
  - Requirements Satisfaction
- Web Demo

General Structure for QA System



- Different researchers are responsible for different modules
- One module's output pass to the other module
- Modules share NLP tools, external resources or machine learning techniques

Problems in General Structure

- Frequent Change of NLP tools
- QA Module Integration
- Consecutive Module Evaluation
- Web-related Considerations

Problem 1 -- Frequent Changing of NLP tools

- Many choices of NLP tools

POS Tagging	Brill's TBL Tagger / Upenn MXPOST Tagger / fnTBL tagger / TiMBL tagger / Edinburgh C&C tools / …
Named Entity Recognition	Lingpipe / Edinburgh C&C tools / Irene's / …
Coreference Resolution	Lingpipe / …
Text Chunking	Abney's Chunker / Edinburgh C&C tools / BaseNP Chunker / …
Parsing	Stanford Parser / Collins Parser / Charniak Parser / CCG Parser / MiniPar / …
Semantic Analysis	…
Reasoning	…

Problem 1 -- Frequent Changing of NLP tools

- One development circle
  - Decide which kind of analysis to use
    - NP chunking or full chunking?
    - PCFG-based Syntactic parsing or dependency parsing?
  - Evaluate NLP tool separately
    - Accuracy vs. Running time
    - Robustness
  - Test various combinations
    - Contributions to the overall performance

→ How to easily and quickly change NLP tools?

Problem 2 – QA Module Integration

- Each Current Module will be enhanced
- New Module will be added into the system
- Different modules are chosen for different processing
  - Web-based passage retrieval module
  - Corpus-based passage retrieval module
  - Answer validation module

→ **How to minimize the effort for module integration?**

Problem 3 – Consecutive Module Evaluation

- Easy to evaluate module separately
  - Question Processing Module
    - Experimental data for question classification
  - Passage Retrieval Module
    - Document Ranking Task in TREC QA
  - Answer Extraction Module
    - generate proper sentences set by TREC judgment file
- Enhance module based on separate evaluation
- Difficult to test the effectiveness of a module on the whole system

→ **How to make a consecutive module evaluation?**

Problem 4 – Web-related Considerations

- Response Time
- Client-Server Communication
- Thread Synchronization
- System Logging
- User Information Backup
  - User address, User request, …

→ **How to cope with all web-related aspects?**

Framework-based Online QA System

- Framework
  - define an overall structure for the system
  - consider all of the aspects
    - not directly related to language processing module
- Functions
  - Provide well defined interfaces for modules and tools
  - Manage collaborations of modules
  - Enable consecutive module evaluation
  - Handle all web-related aspects

Requirements of Framework 1

- Modularity
  - Minimize dependencies
    - between modules
    - between modules and framework
      - Framework won't be changed any more once it is built
    - To minimize effects of code modification
  - Interaction only over a small interface

- Flexibility
  - Dynamically load modules into framework
  - Allowing to plug in/out arbitrary modules
  - Allowing to pass data in any format between modules

Requirements of Framework 2

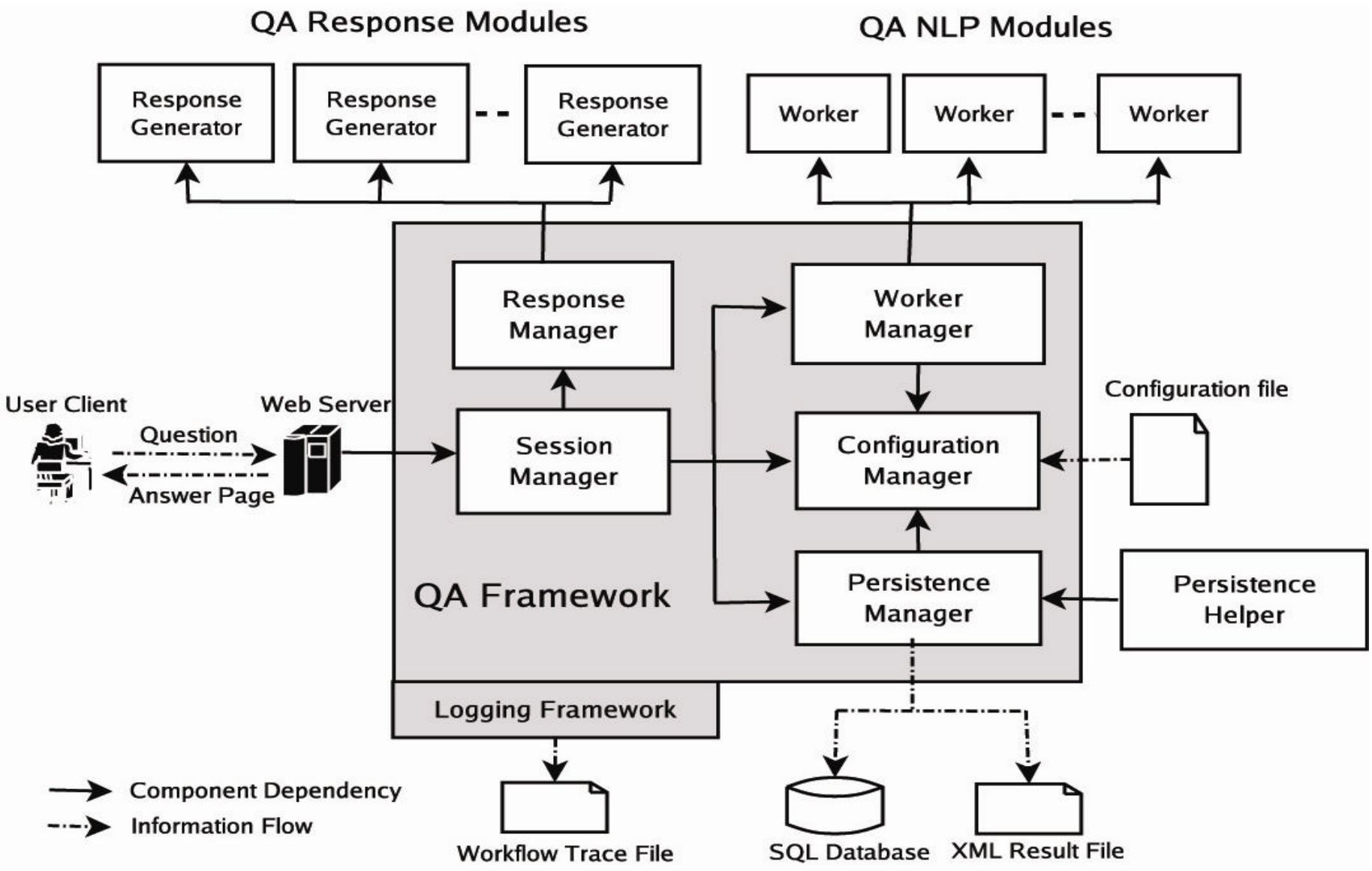
- Configurability – for system setting
  - Not distributed across the whole source codes
  - Exposed to the users
  - Access a readable and editable configuration file
  - Avoids recompiling
- Scalability
  - Adjustable with respect to hardware
  - How many user can be served in parallel?
    - Max. number of user requests
    - Max. ability of resource consumption
      - CPU, working memory

Requirements of Framework 3

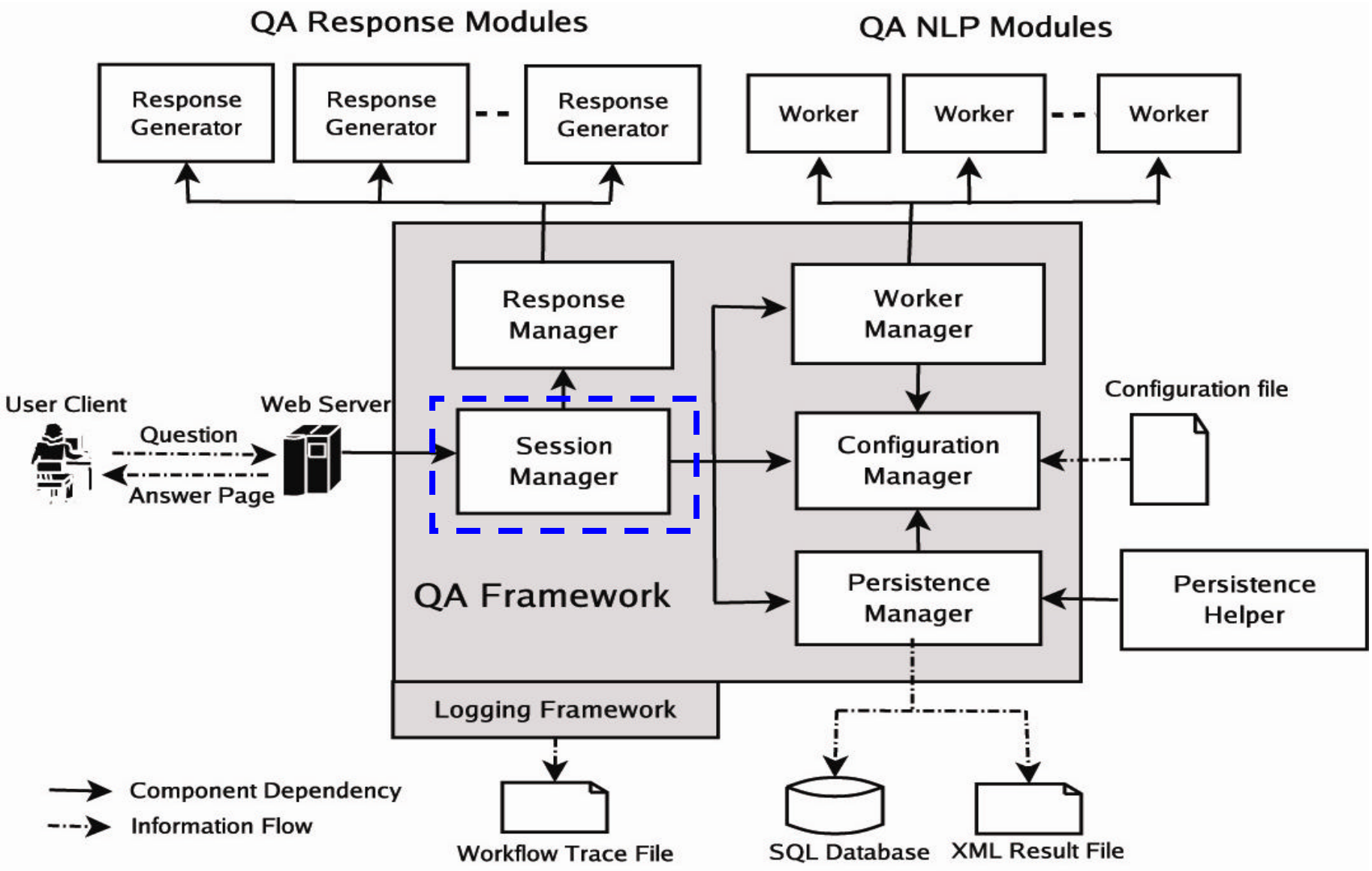
- Initialization
  - Long initialization time of Modules
  - Separate module work into 2 phases
    - time-uncritical (one time) initialization phase
    - time-critical (frequent) working phase
  - hold modules in working memory

- Synchronization
  - Web-Apps work in multi-thread mode
  - Shared objects have to be synchronized
  - One thread per user request

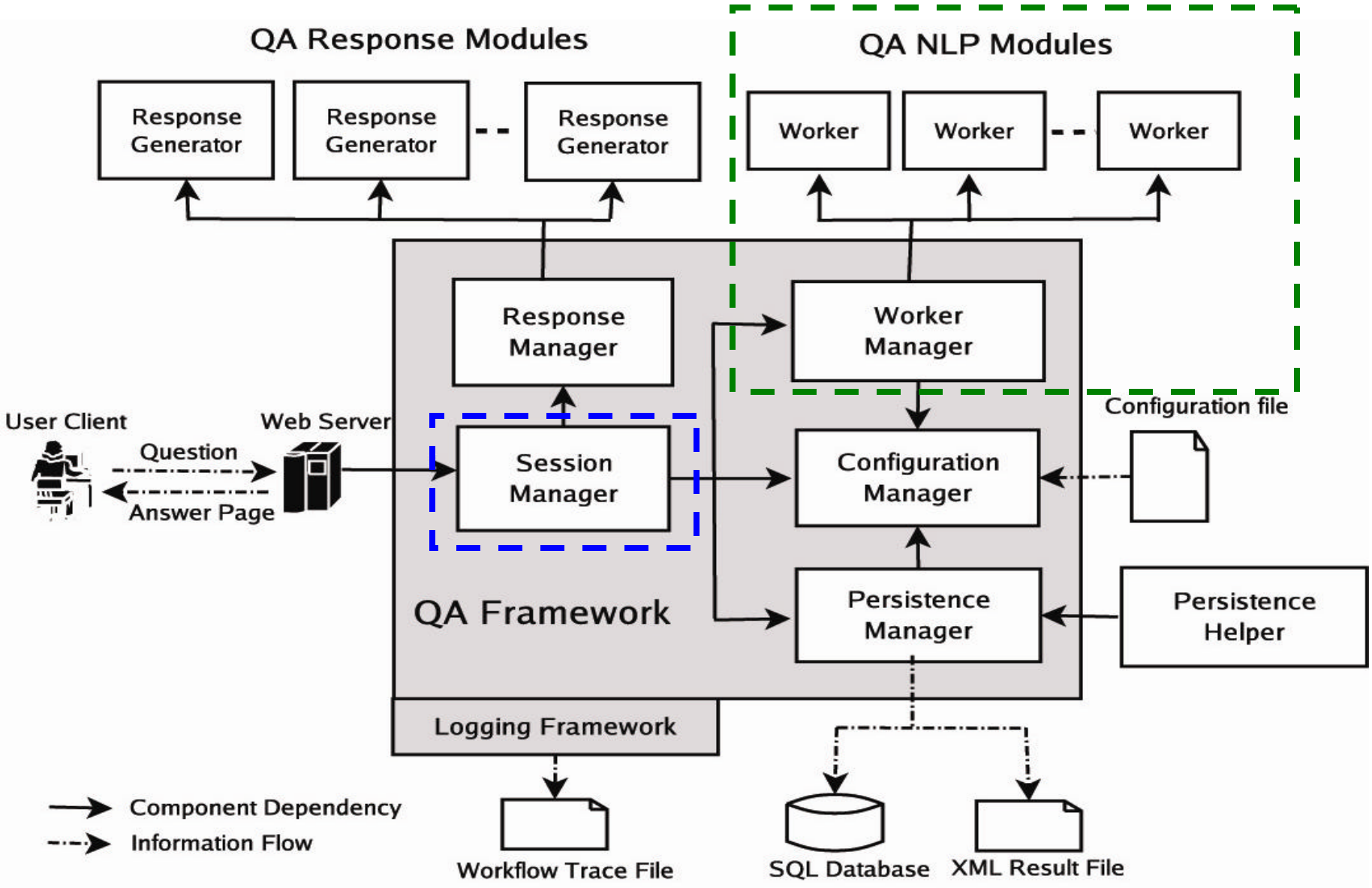
Framework


QA Response Modules
QA NLP Modules
Response Generator
Response Generator
Response Generator
Worker
Worker
Worker
Response Manager
Worker Manager
User Client
Web Server
Question
Answer Page
Session Manager
Configuration Manager
Configuration file
QA Framework
Persistence Manager
Persistence Helper
Logging Framework
Component Dependency
Information Flow
Workflow Trace File
SQL Database
XML Result File

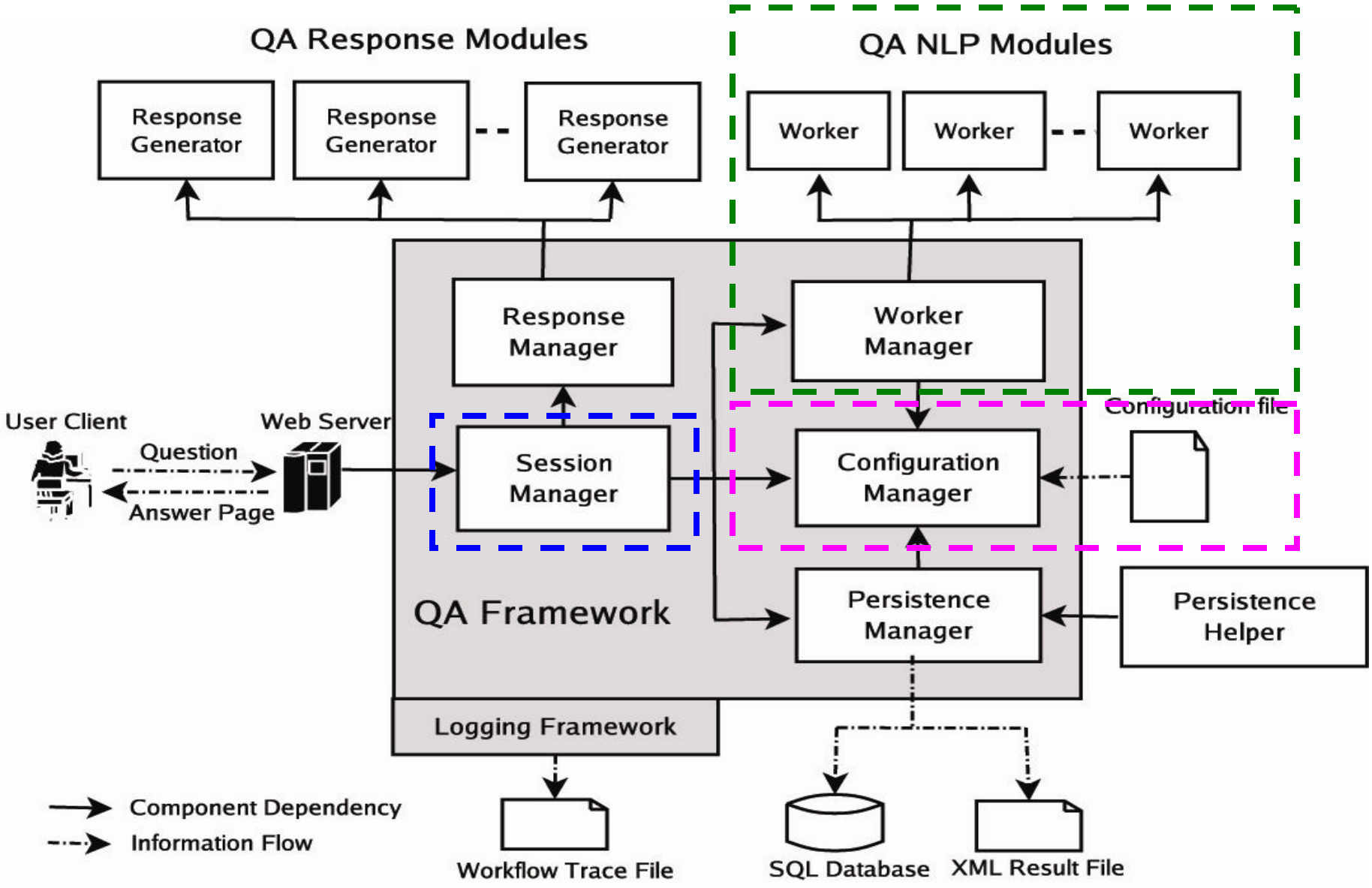
Framework


QA Response Modules
QA NLP Modules
Response Generator
Response Generator
Response Generator
Worker
Worker
Worker
Response Manager
Worker Manager
User Client
Web Server
Question
Answer Page
Session Manager
Configuration Manager
Configuration file
QA Framework
Persistence Manager
Persistence Helper
Logging Framework
Component Dependency
Information Flow
Workflow Trace File
SQL Database
XML Result File

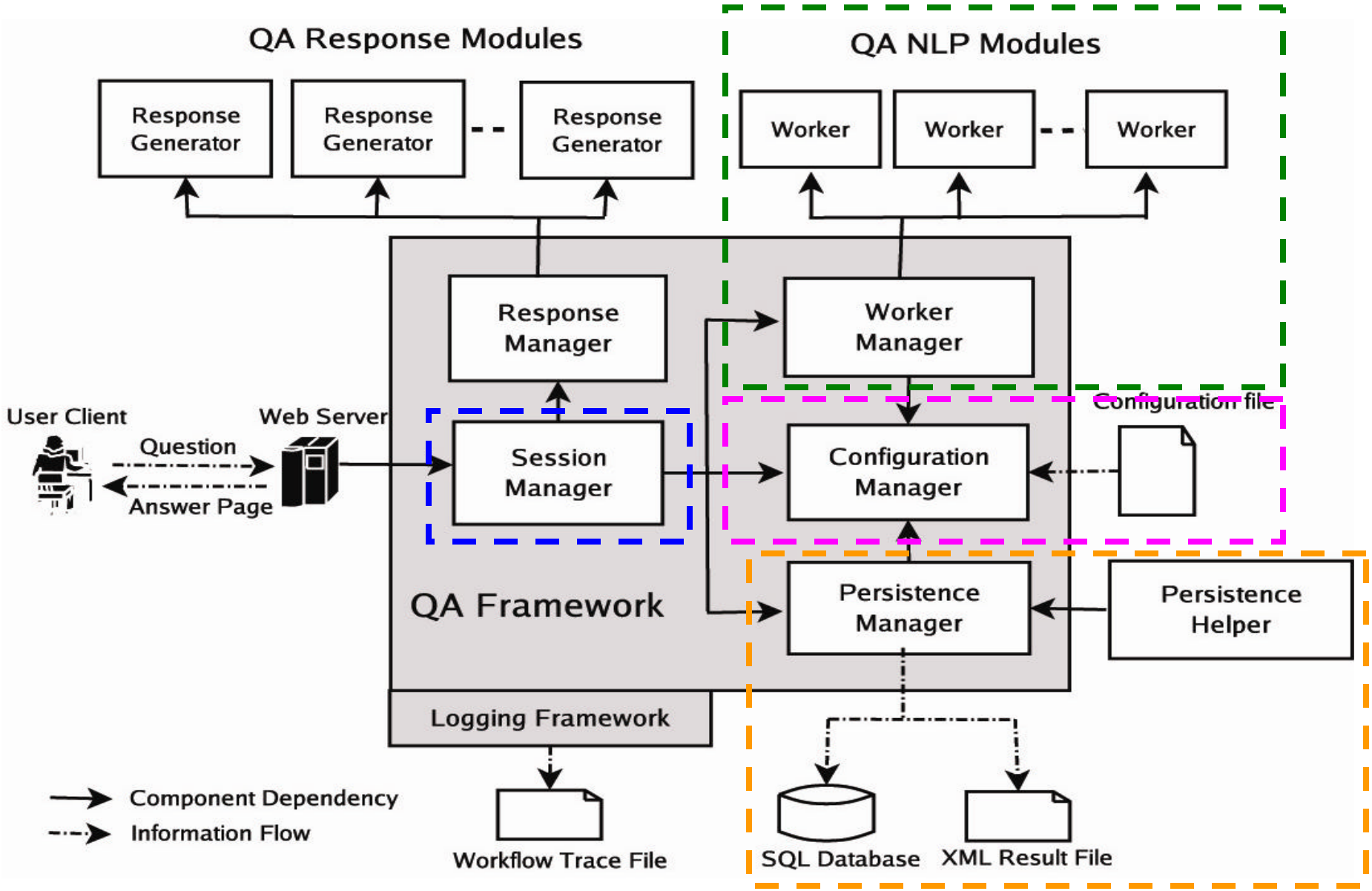
Framework


QA Response Modules
Response Generator
Response Generator
Response Generator
QA NLP Modules
Worker
Worker
Worker
Response Manager
Worker Manager
Configuration file
User Client
Web Server
Question
Answer Page
Session Manager
Configuration Manager
QA Framework
Persistence Manager
Persistence Helper
Logging Framework
Component Dependency
Information Flow
Workflow Trace File
SQL Database
XML Result File

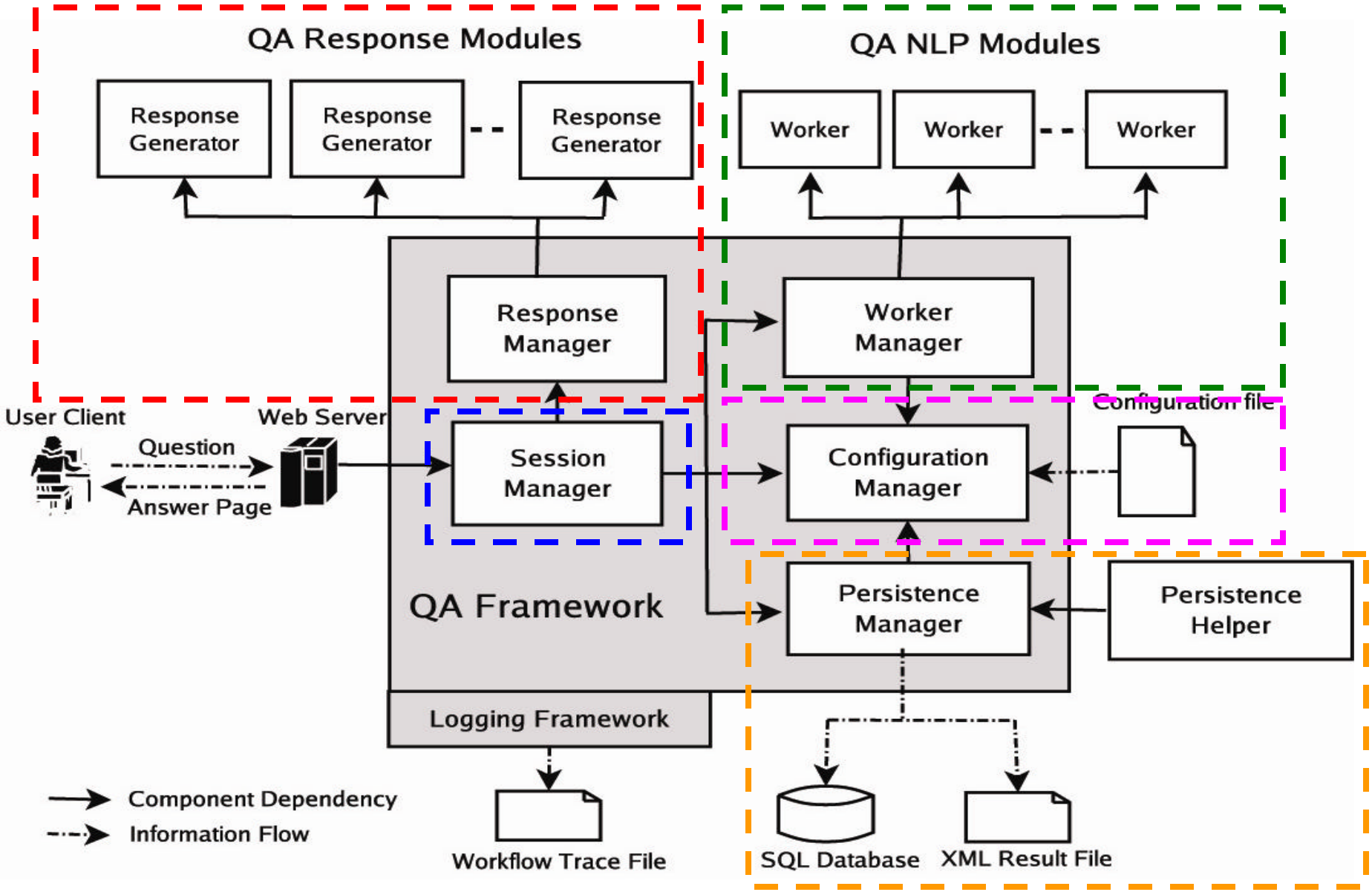
Framework


QA Response Modules
Response Generator
Response Generator
Response Generator
QA NLP Modules
Worker
Worker
Worker
Response Manager
Worker Manager
User Client
Web Server
Question
Answer Page
Session Manager
Configuration Manager
Configuration file
QA Framework
Persistence Manager
Persistence Helper
Logging Framework
Component Dependency
Information Flow
Workflow Trace File
SQL Database
XML Result File

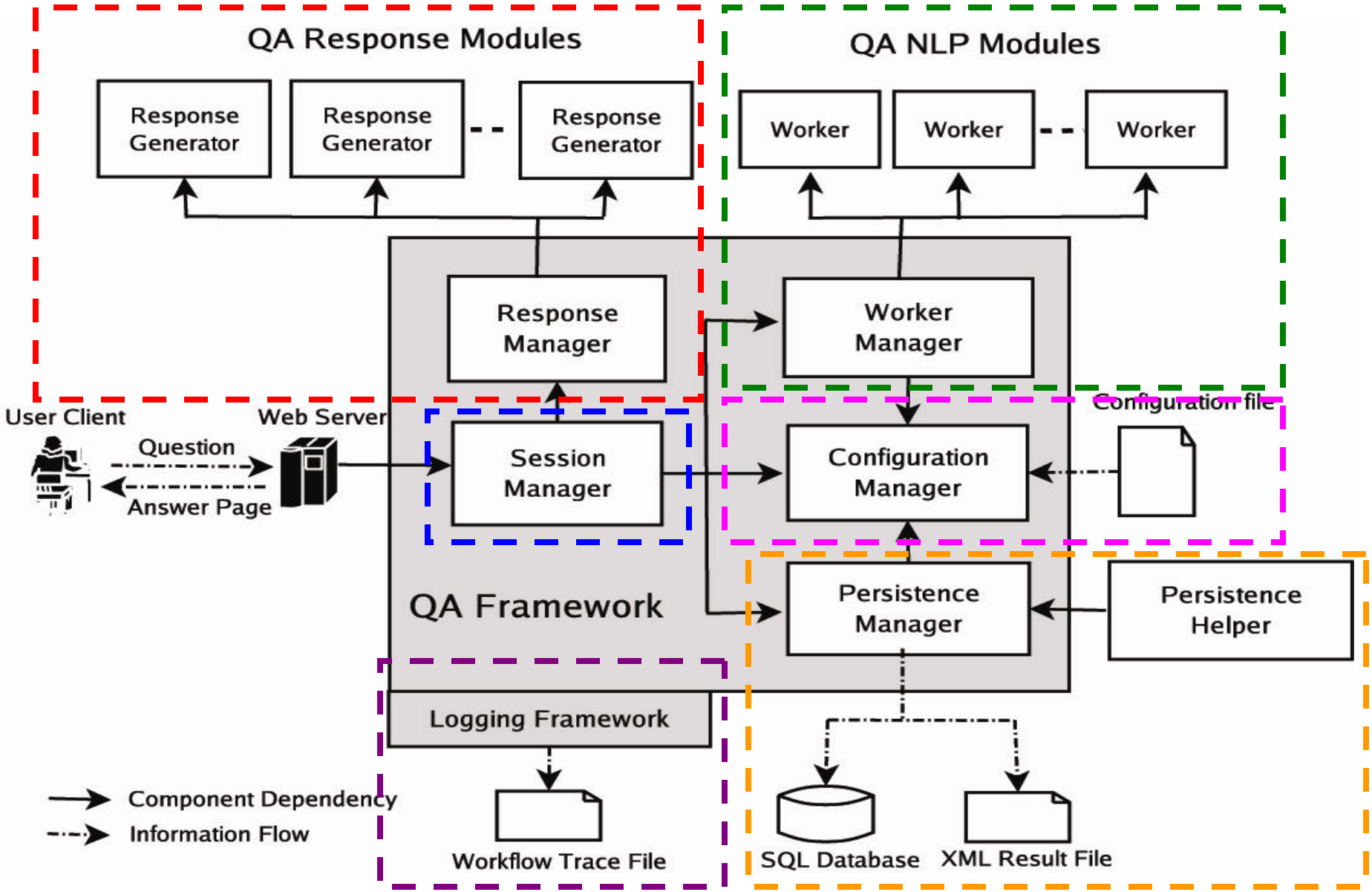
Framework


QA Response Modules
QA NLP Modules
Response Generator
Response Generator
Response Generator
Worker
Worker
Worker
Response Manager
Worker Manager
User Client
Web Server
Question
Answer Page
Session Manager
Configuration Manager
Configuration file
QA Framework
Persistence Manager
Persistence Helper
Logging Framework
Component Dependency
Information Flow
Workflow Trace File
SQL Database
XML Result File

Framework

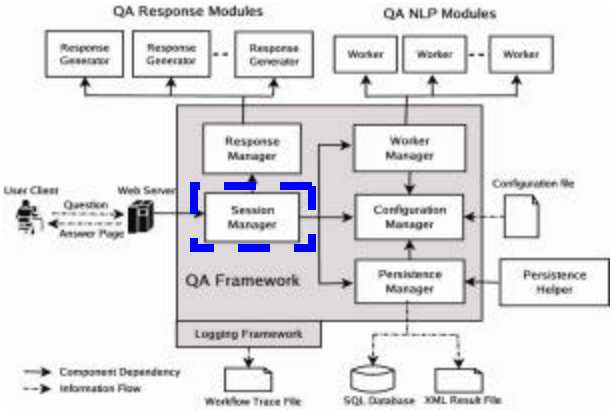

QA Response Modules
Response Generator
Response Generator
Response Generator
QA NLP Modules
Worker
Worker
Worker
Response Manager
Worker Manager
User Client
Web Server
Question
Answer Page
Session Manager
Configuration Manager
Configuration file
QA Framework
Persistence Manager
Persistence Helper
Logging Framework
Component Dependency
Information Flow
Workflow Trace File
SQL Database
XML Result File

Framework


QA Response Modules
Response Generator
Response Generator
Response Generator
QA NLP Modules
Worker
Worker
Worker
Response Manager
Worker Manager
User Client
Web Server
Question
Answer Page
Session Manager
Configuration Manager
Configuration file
QA Framework
Persistence Manager
Persistence Helper
Logging Framework
Component Dependency
Information Flow
Workflow Trace File
SQL Database
XML Result File

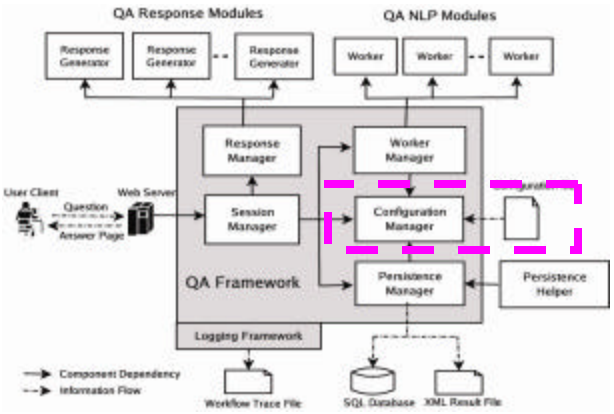
Session Manager

- Java servlet class
- Accept user question
- return answer to user
- Control all activities on the top level
- Initialization Stage
  - Instantiate *Configuration Manager*, *Persistence Manager*, *Response Manager*, *Logger Manager*
  - Build a Worker Manager Pool
    - Instantiate *n Worker Managers*
    - *Put them into Work Manager Pool*
- Working Stage
  - Get a work manager from pool to process a question



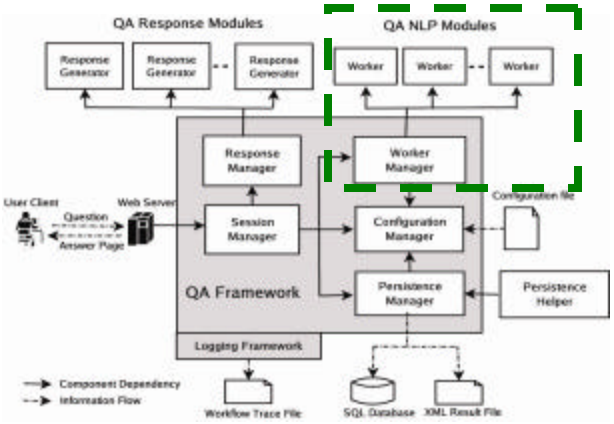
Configuration Manager

- Read a XML configuration file
- Provide Session Manager
  - Maximum number of the users
- Provide Work Manager
  - Which processing modules to use
  - Information for NLP tools
- Provide Persistence Manager
  - Database Information



Worker Manager

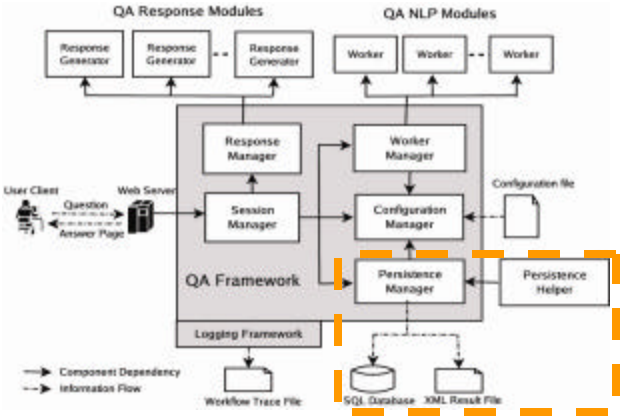
- Instantiate the selected modules
  - Question Processing Module
  - Passage Retrieval Module
  - Answer Extraction Module
  - …



- Call the modules to process user questions and extract answer
- A standardized interface ***Worker***
- Hold in working memory once system runs

Persistence Manager

- Gather data
  - Question
  - retrieved sentences, extracted answers
  - Running time for each module



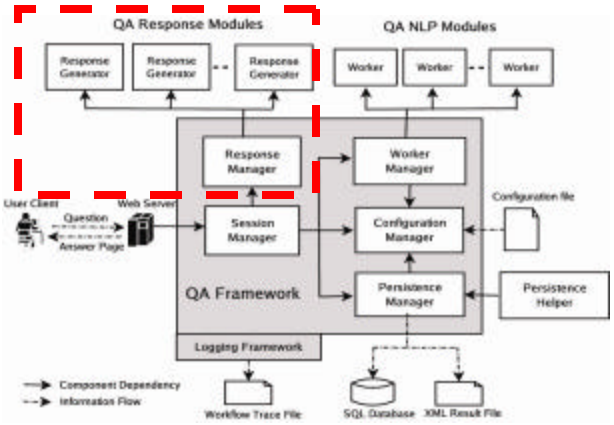
- Offline Evaluation
  - accuracy and running time
  - single modules / module combinations

- Question Corpus
  - Further enhance question processing module

- Logging – to Database / XML file

Response Manager

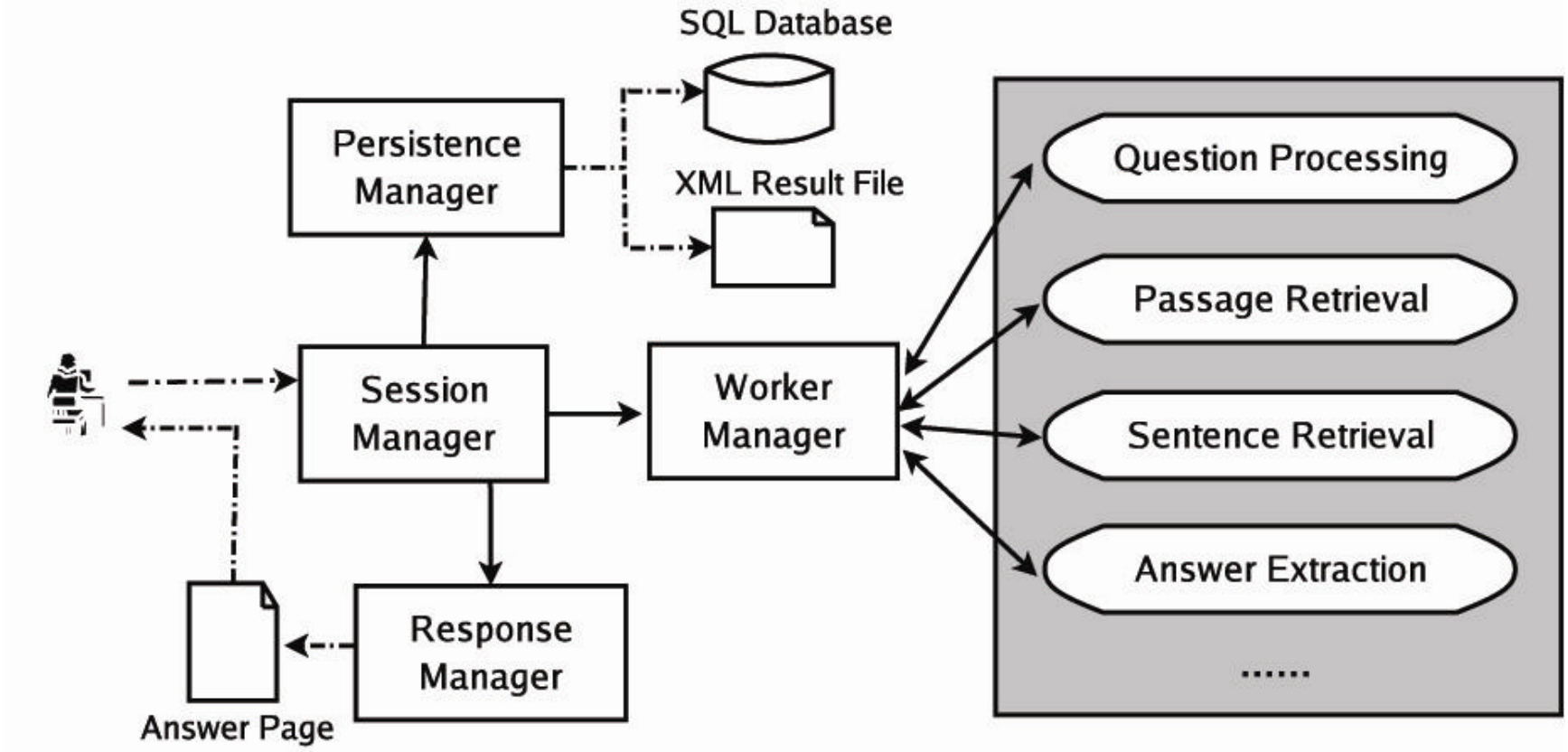
- Choose an output format
  - the specifics of requesting client



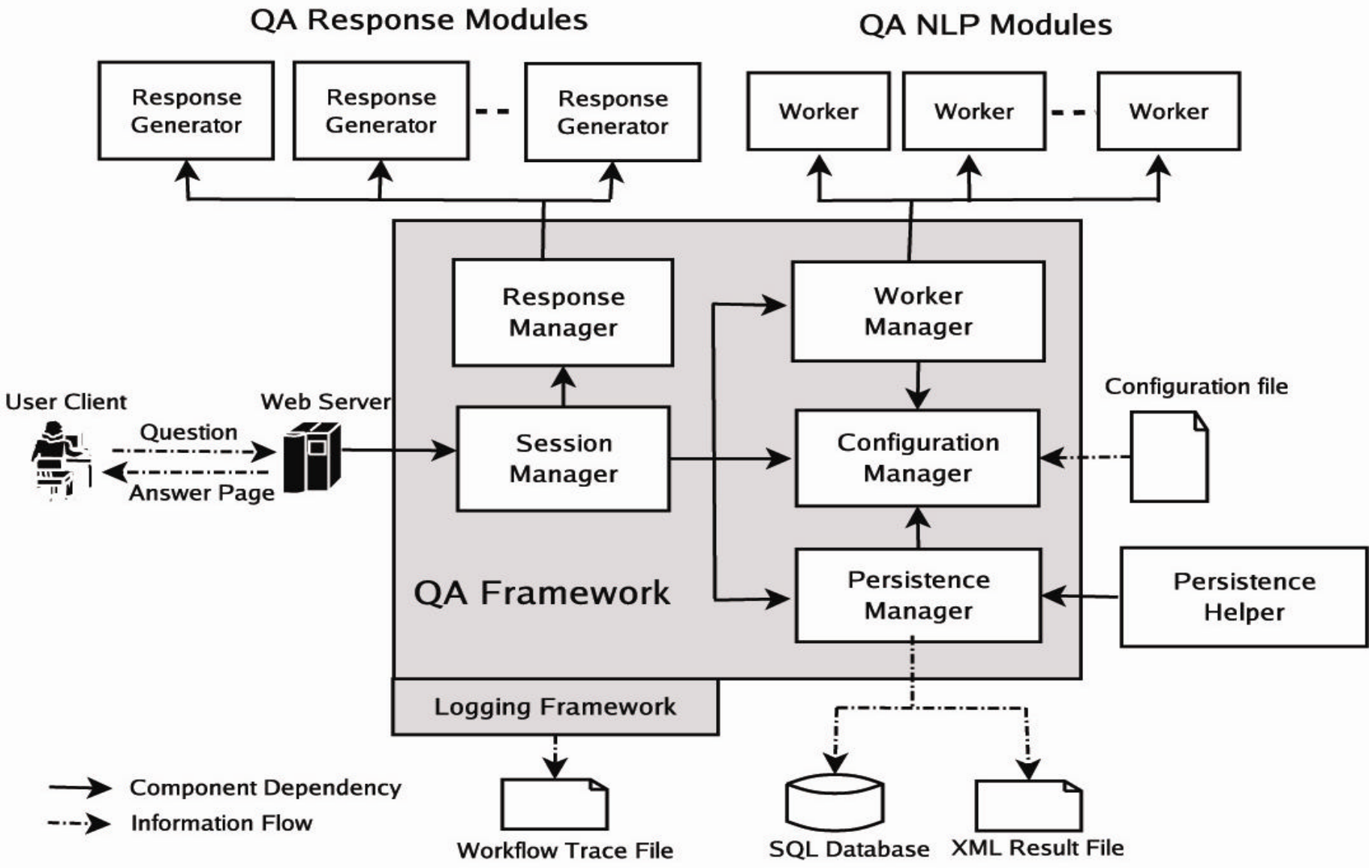
- Different output formats
  - Specified by the respective Response Generator module
  - HTML pages for normal PC clients
  - WML pages for small screened devices (PDA)

Collaboration of Modules

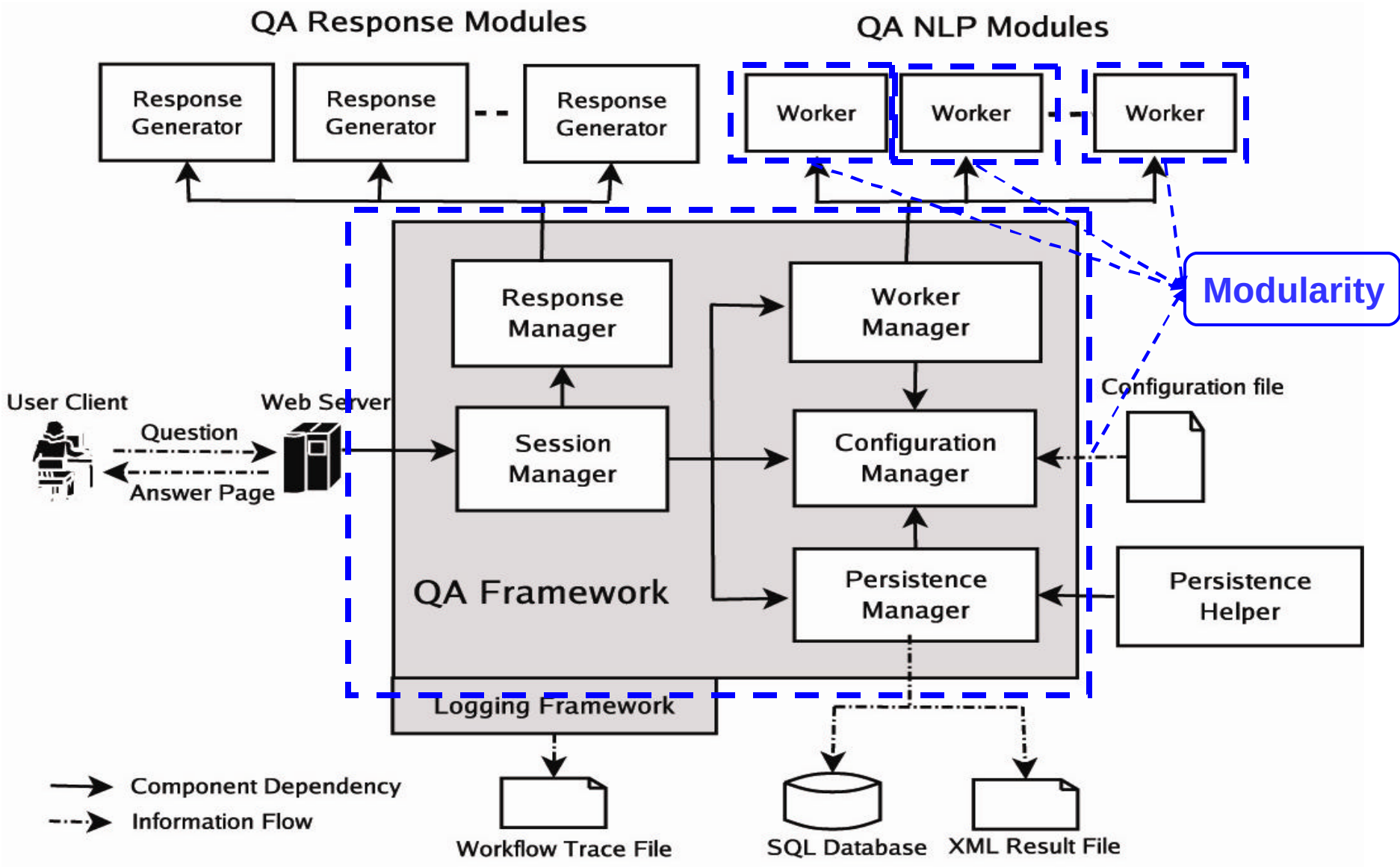
- datasheet object
  - Instantiated by Session Manager
  - Exchange information between modules



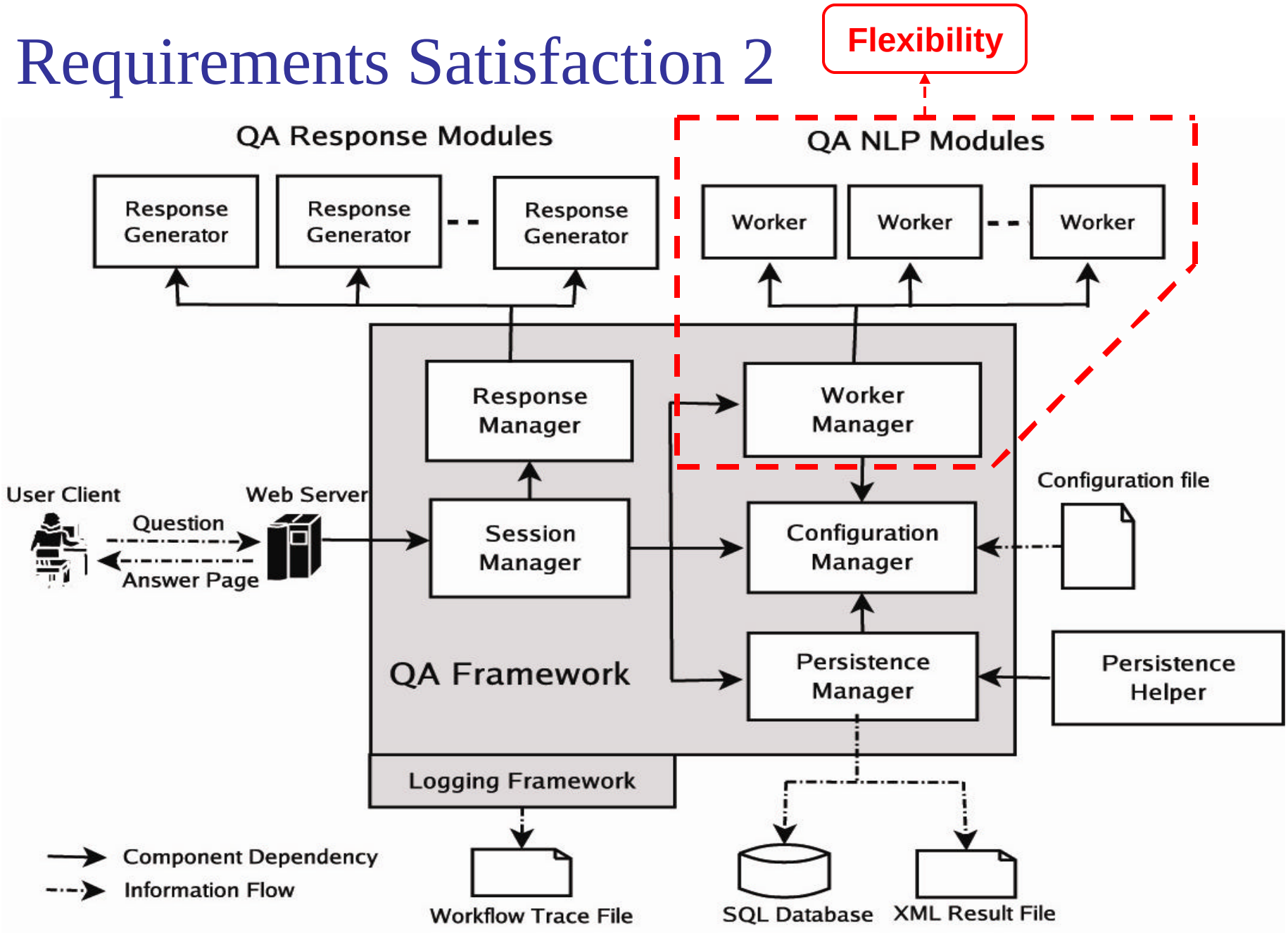
Requirements Satisfaction



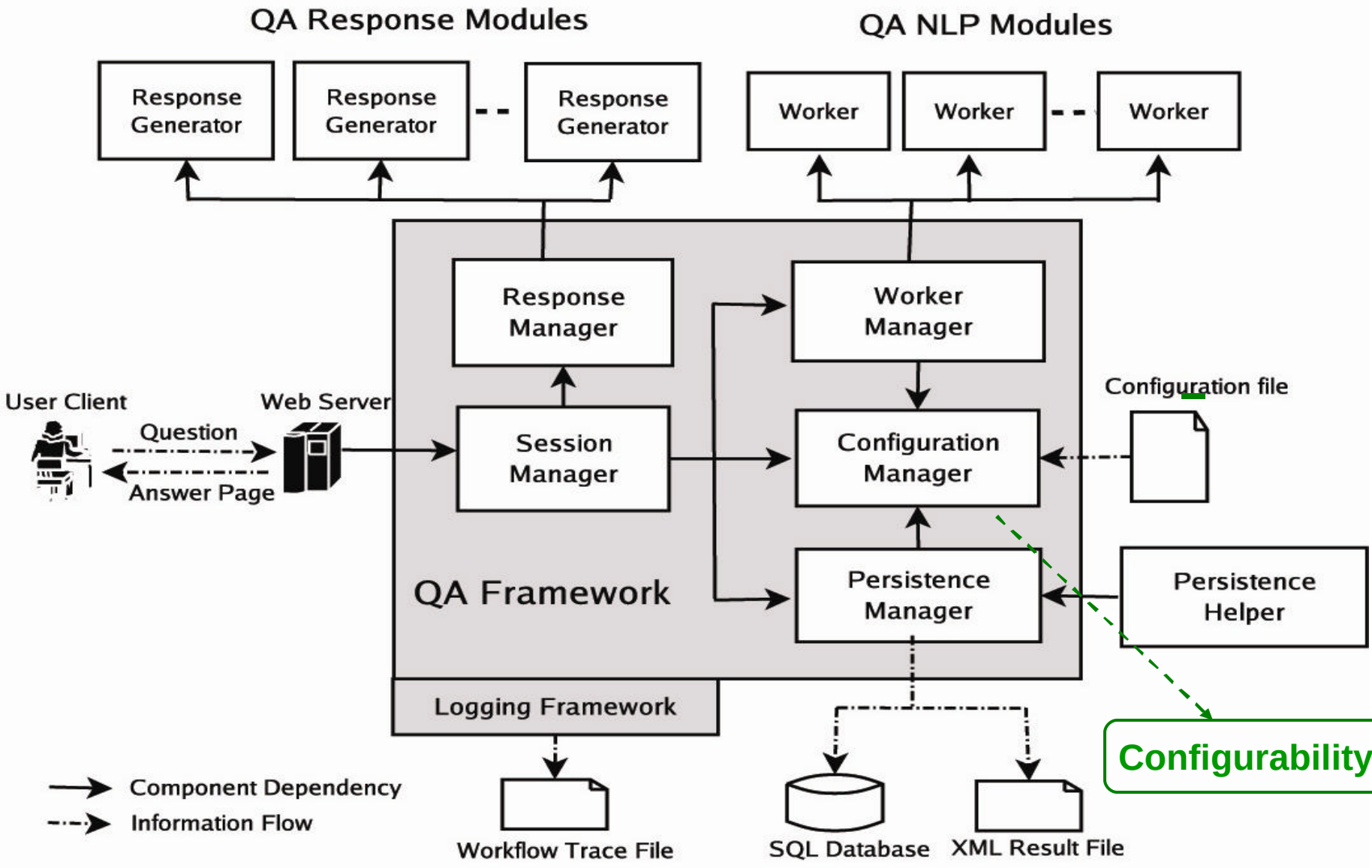
Requirements Satisfaction 1



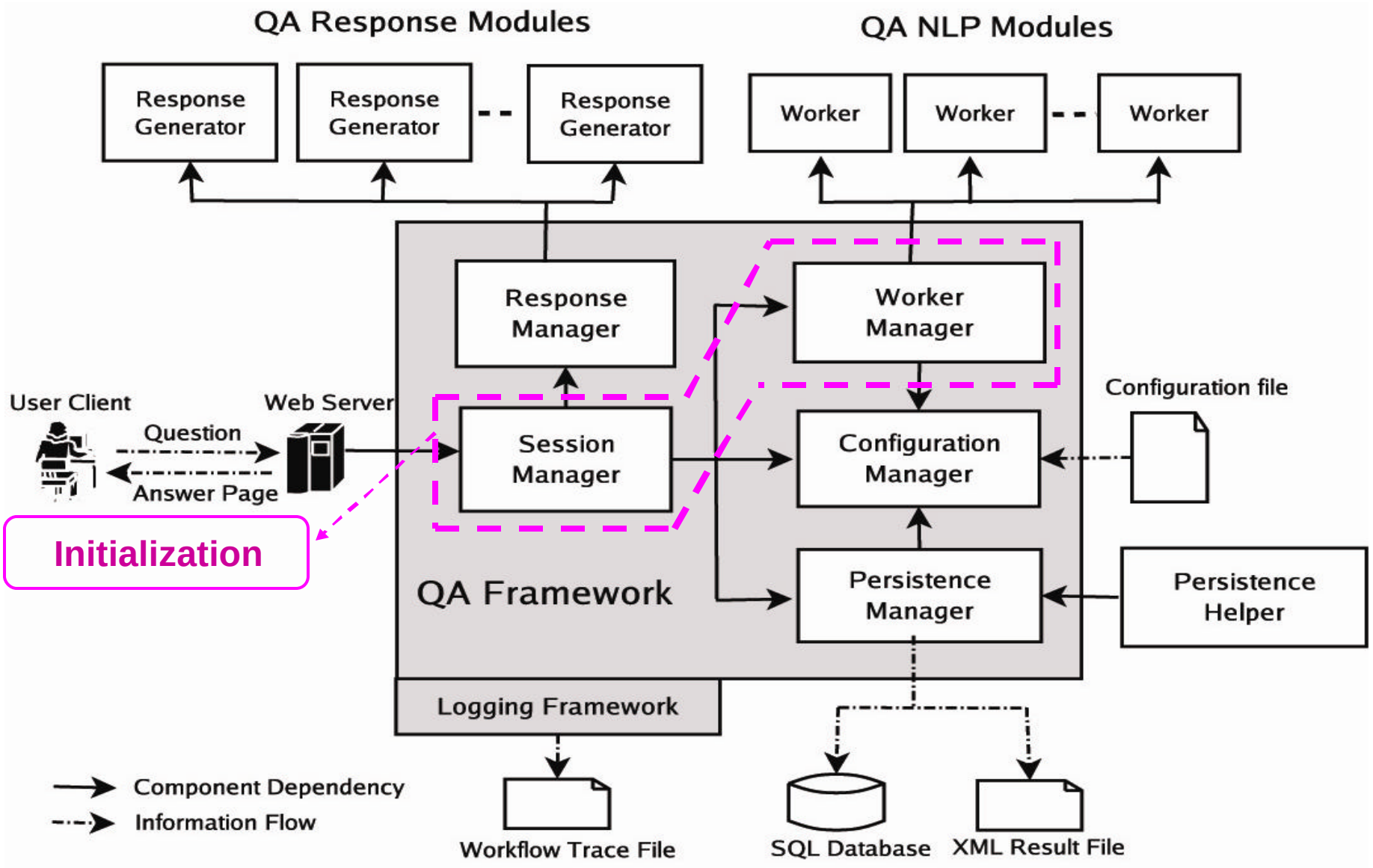
Requirements Satisfaction 2



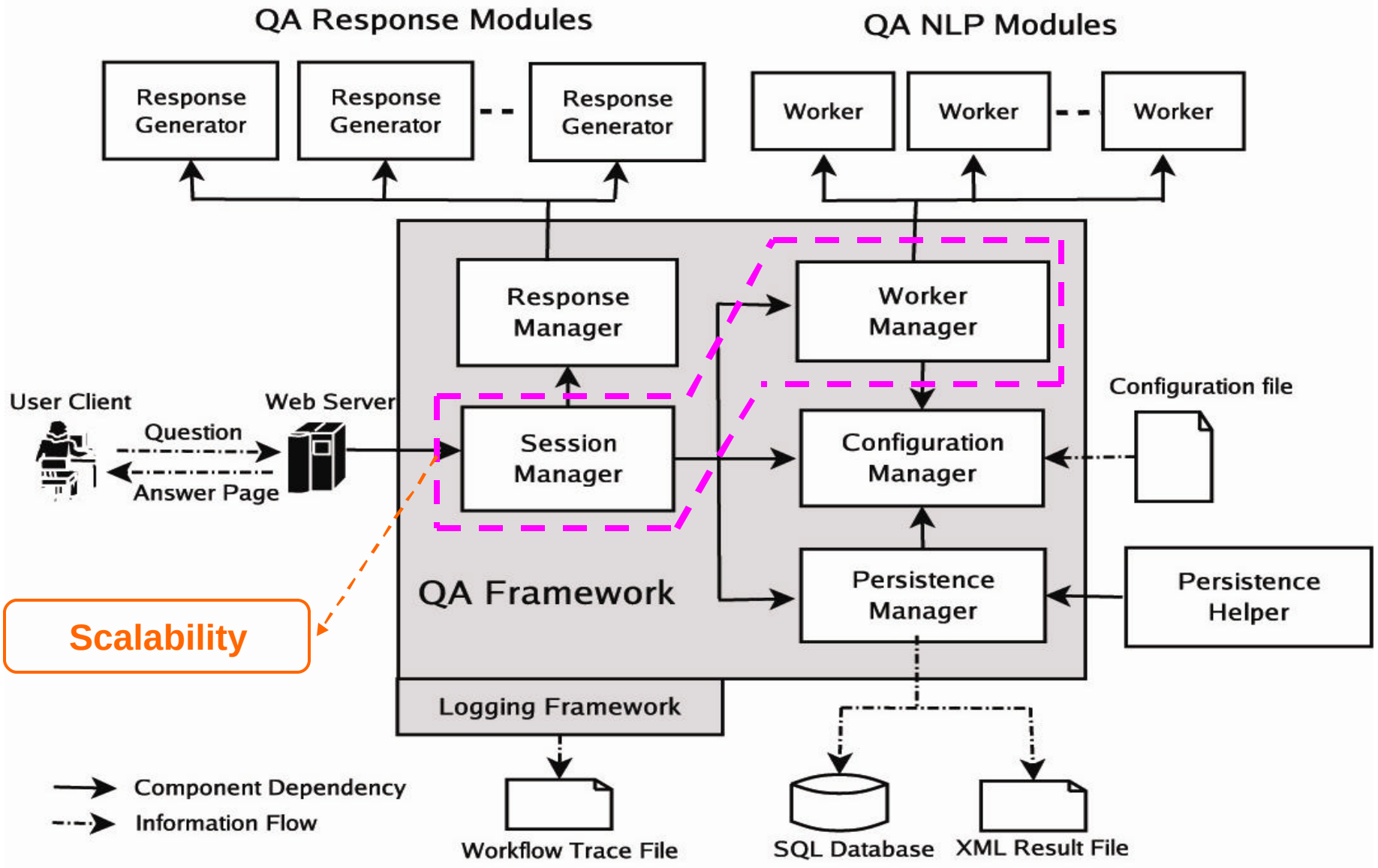
Requirements Satisfaction 3



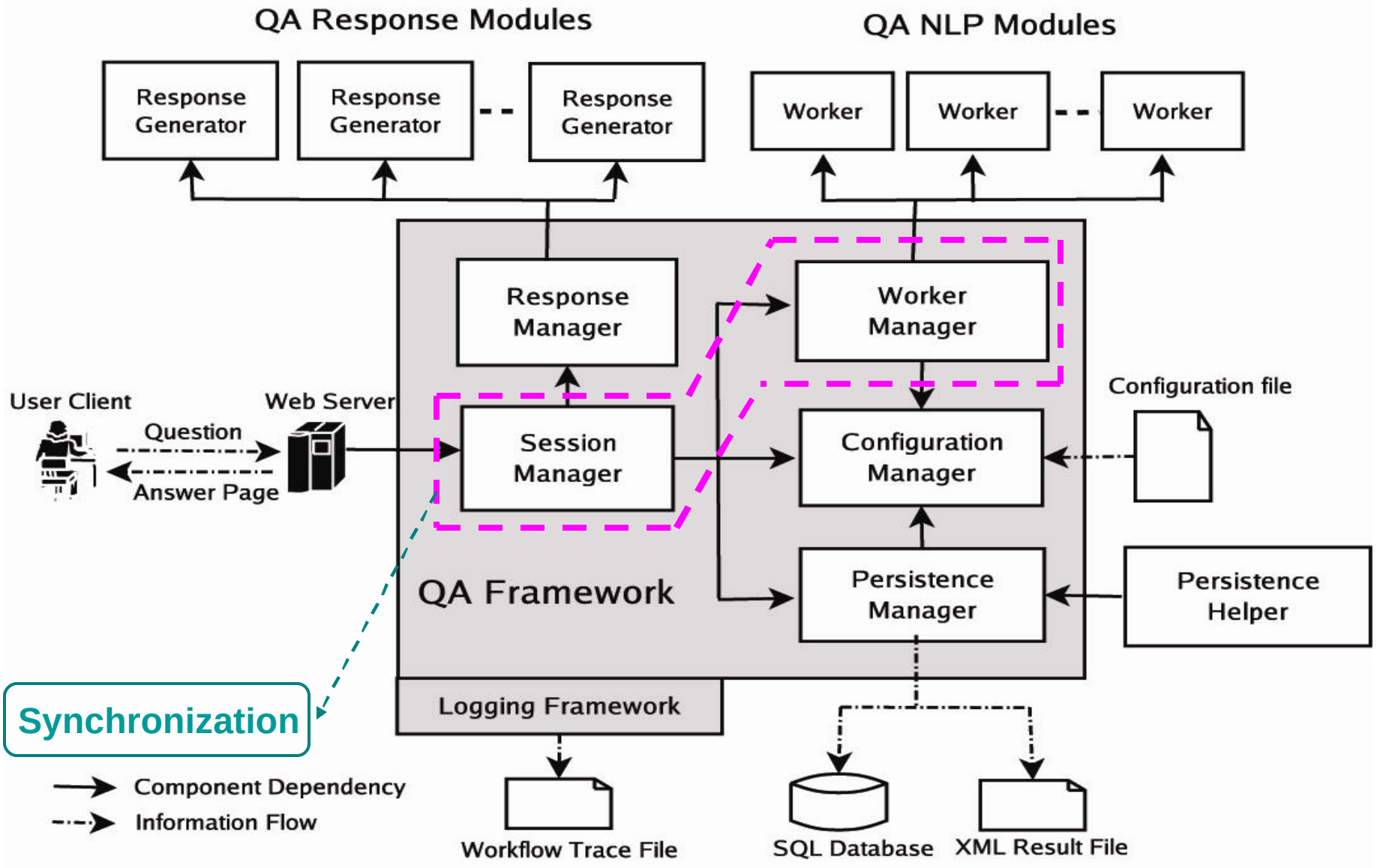
Requirements Satisfaction 4



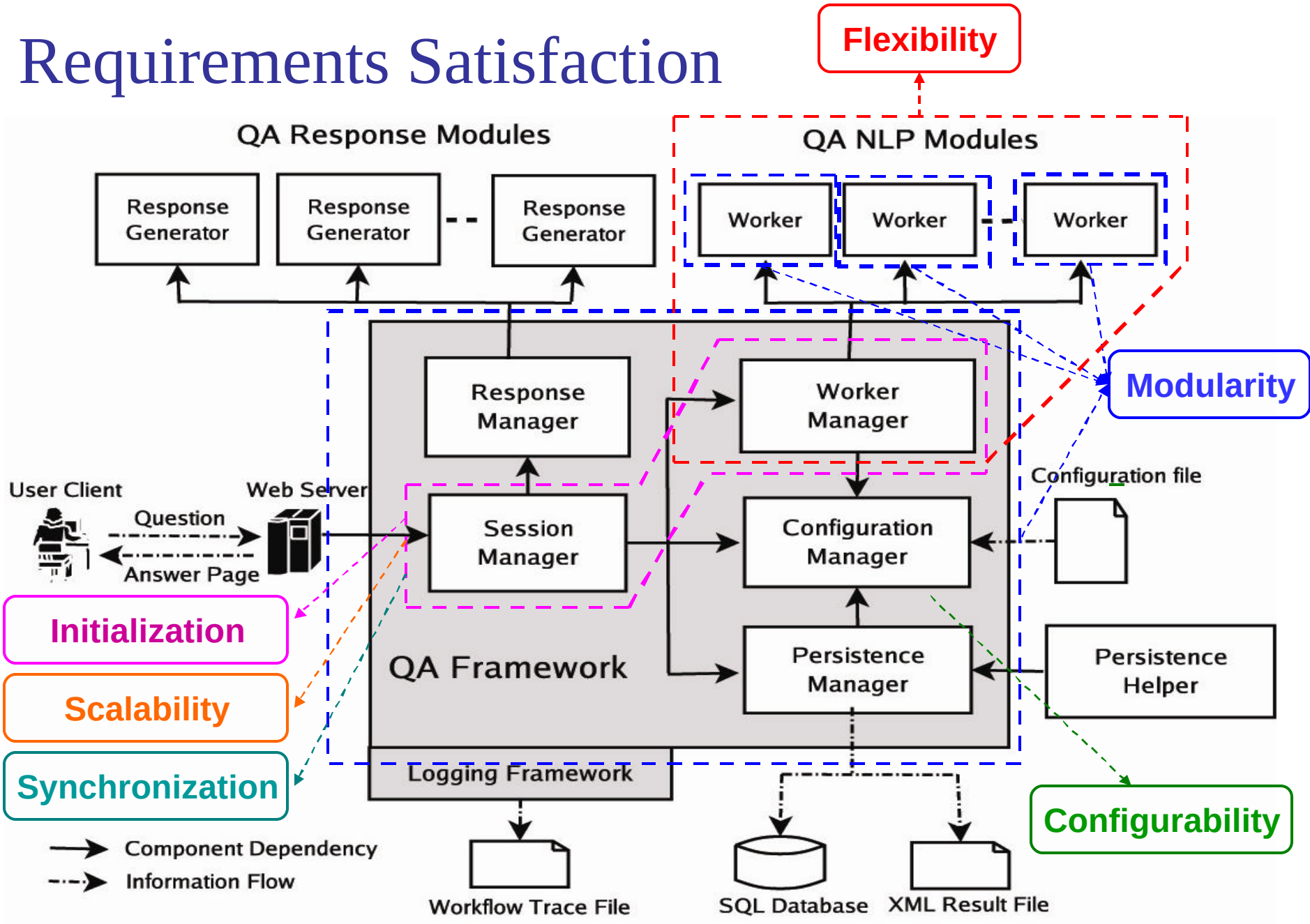
Requirements Satisfaction 5



Requirements Satisfaction 6



Requirements Satisfaction



Conclusion

- Framework for online QA system
  - Web-related aspects
  - Module integration
  - Module collaboration
  - Consecutive module evaluation

- Release QA researchers from all non-NLP related tasks
- Platform for researchers working on QA together

Demo …