TABLE OF CONTENTS

1. Introduction

1.1. Document overview

This document describes the architecture of ESPRESSO system.
It describes:

- A general description of the system
- The logical architecture of software, the layers and top-level components
- The physical architecture of the hardware on which runs the software
- The justification of technical choices made
- The traceability between the architecture and the system requirements.

1.2. Abbreviations and Glossary

1.2.1. Abbreviations

Add here abbreviations
COTS: Components Off the Shelf (software industry acronym)
OTSS: Off The Shelf Software (FDA acronym)
SOUP: Software Of Unknown Provenance (IEC 62304 acronym)
VTI: Virtual Table Interface

1.2.2. Glossary

Solid Server - a server conforming to the Solid specification
Pod - Personal Online Datastore as part of a Solid Server
GaianDB -
Gaian DB node -
Brewmaster -
Barista -
CoffeeFilter -
Pod Index - Index that contains information about the data stored in the pod
Meta Index -
Searchable data -
User query - what user enters into the search form. Can be a keyword, can be a SPARQL query.

1.3. References

1.3.1. Project References

#	Document Identifier	Document Title
[R1]	ID	

1.3.2. Standard and regulatory References

#	Document Identifier	Document Title
[STD1]		

1.4. Conventions

2. Architecture

2.1. Architecture overview

The system provides a user with online search functionality over data stored in pods on SOLID Servers. It contains an indexing app that creates indices in the pods and provides a querying interface for keyword search and SPARQL queries execution. It propagates the query through the network of GaianDB nodes that are connected to the SOLID Servers and retrieve data to the user.

2.2. Physical architecture overview

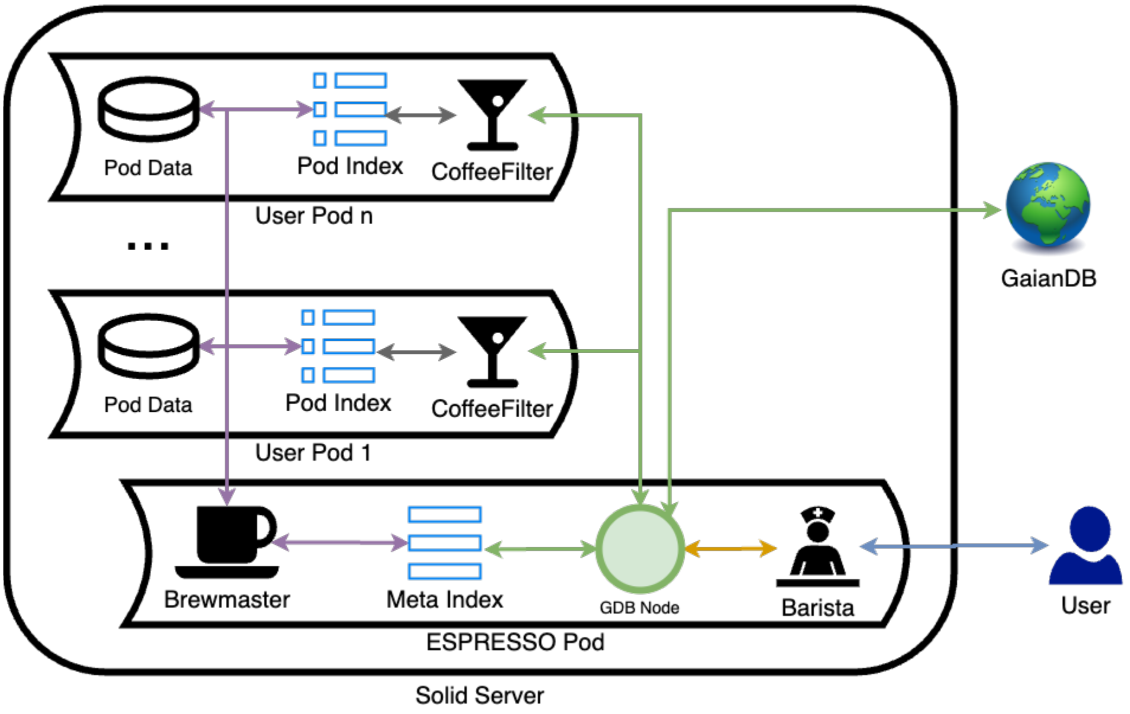
2.2.1. Hardware Component 1 description

2.2.2. Hardware Component 2 description

2.2.3. Hardware Component 3 description

2.3. Logical architecture overview

The Brewmaster connects to the pod and produces a local index that is stored in the pod. The CoffeeFilter monitors the incoming requests to see the index and returns the relevant part of the index based on the credentials provided. Barista is the user interface - user provides it with their query and credentials, and Barista retrieves the results.

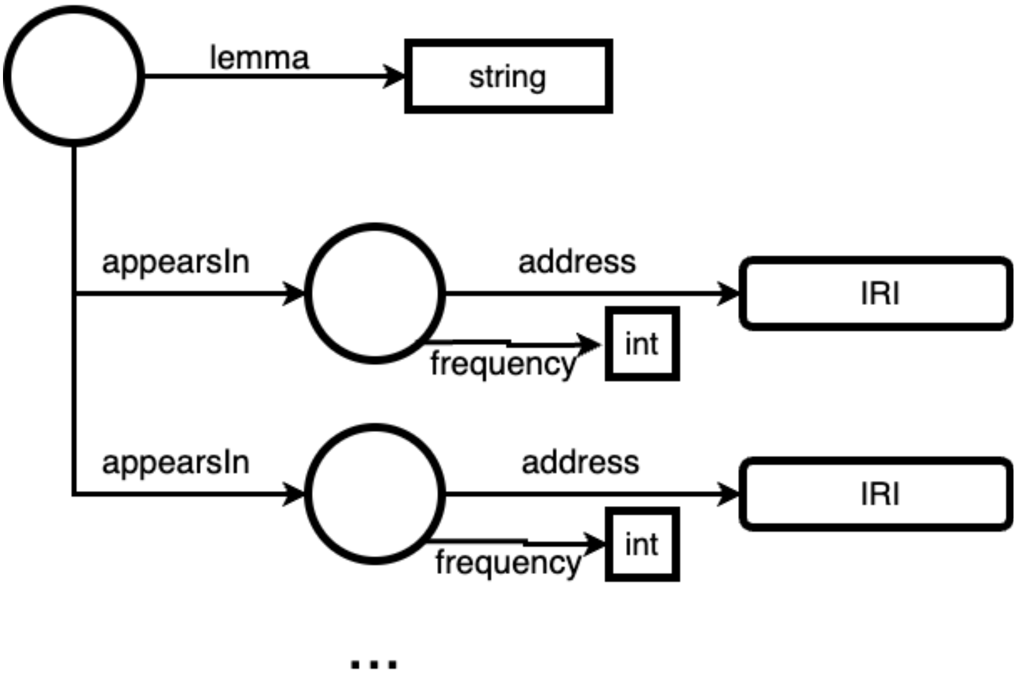


2.3.1. The Brewmaster description

Brewmaster creates or updates the Pod Indices and the Meta Index.

2.3.1.1. The Pod Index description

The Pod Index is a serialisation of an pdf-graph of with the following structure:



2.3.2. The CoffeeFilter description

CoffeeeFilter reacts to the requests to access the Pod Index and returns only the relevant part.

2.3.3. The Barista description

Barista is the user interface app for the search. It accepts users credentials and query, and turns it into a specially formed GaianDB query. Then it processes the Gaian DB response and formats it for the user.

2.4. Software SOUP

Gaian DB - A light-weight data-federation technology built on Apache Derby (https://github.com/gaiandb/gaiandb). ESPRESSO uses a custom built version of Gaian DB with custom VTIs.
Community Solid Server - The Community Solid Server is open software that is an open and modular implementation of the Solid specifications. Used in ESPRESSO for setting up servers for system testing and development (https://github.com/CommunitySolidServer).

3. Dynamic behaviour of architecture

The ESPRESSO Search system consists of two main parts - Indexing and Search.

3.1. Adding a server to the ESPRESSO system

When a Solid server is added to the ESPRESSO System the following happens:

1. An ESPRESSO pod is created on the server that contains Brewmaster app, Barista app, a Gaian DB node, and the Meta index.
2. The Gaian DB node is connected to other nodes in the ESPRESSO system.
3. The Brewmaster app is configured for the server.
4. Brewmaster does the initial indexing - creates pod indices in the user pods and the Meta index.
5. CoffeeFilter is installed to every pod.

3.2. Updating the indices (by request)

For the Solid server connected to the ESPRESSO system, whenever there is a change in the searchable data the following happens:

1. A Brewmaster update request is formed and sent to the Brewmaster by the server
2. The Brewmaster connects to the pod where the change occurs and updates the Pod index.
3. The Brewmaster updates the Meta index.

3.3. Updating the indices (automatic)

Periodically (as is specified during the configuration) the Brewmaster reindexes the server as if it was just connected.

3.4. Search

The search happens as follows:

1. (optional) User authenticates themselves to the Barista app.
2. User provides a query to Barista.
3. Barista produces a formatted query based on the user query and the credentials provided.
4. The formatted query is sent to the Gaian DB node.
5. The query is propagated through the Gaian DB network.
6. When the query arrives to a Gaian DB node associated with a solid server the following happens:
    1. The node forms a logical table based on the information in the Meta index
    2. The Pod indices are accessed and the information provided by the CoffeeFilters in the pods is added to the logical table
    3. The logical table is queried with the query and the result is sent back through the Gaian DB network
7. Barista collects the result from Gaian DB, formats it and presents to the user.

4. Justification of architecture

4.1. System architecture capabilities

4.2. Network architecture capabilities

4.3. Risk analysis outputs

4.4. Human factors engineering outputs

4.5. SOUP integration

5. Requirements traceability

Requirement	Component	Comment
REQ-001 Indexing of the pod data	Brewmaster	Brewmaster connects to the pod and produces a local index that is stored in the pod.
REQ-002 Creating Meta Index	Brewmaster	Brewmaster creates and updates the Meta Index
REQ-003 User search query answering	Barista	