D4.1 Specification Report

File name	PuppyIR_D4.1_Spec_Report_0.2.doc
Author(s)	Leif Azzopardi (UGLW) Richard Glassey (UGLW) José Miguel Garrido (Atos) Elena Garrido (Atos)
Work package/task	WP4/Tx.n
Document status	Intermediate draft/ready for approval/final
Version	0.2.1
Contractual delivery date	31/01/2010
Confidentiality	confidential
Keywords	Software specification
Abstract	Specification report for the open source PuppyIR Framework

Table of Contents

Executive Summary

This report details the specification of the open source PuppyIR Framework. This specification is derived from the following deliverables from WP1:

- D1.2 – Agreed User Requirements and Scenarios
- D1.3 – Agreed Technical Requirements

The purpose of this report is to specify the **software architecture** that will guide the forthcoming **design** (D4.2 – Design Report) and **development** (D4.3 – Report on Implementation and Documentation and D4.4 Release of Open Source framework, version 1.0) of the PuppyIR Framework.

The software architecture will be described from a high-level **system context** viewpoint, the various design **considerations** and **constraints** that are relevant to the development of the framework, and by decomposing the architecture into three views, 1) an architecture overview, 2) a functional architecture, and 3) an operational architecture.

1 Introduction

The PuppyIR Project aims to facilitate the creation of child-centered information services, based on the understanding of the behaviour and needs of children [1]. A major component required to achieve this aim is the development of an open source framework that allows systems developers to make new information services by reusing and extending a set of components. The purpose of this document is to provide the specification of the PuppyIR Framework.

- *Provide a general, non-technical description of the PuppyIR software system to be built*
- *Goals of the specification report*
    - *Start from a high-level viewpoint and describe the context of the system*
    - *Raise the considerations and concerns that affect the development*
    - *Explore and develop the software architecture from multiple viewpoints*

1.1 Audience and Scope of Specification

- *Identify the audience (Stakeholders, Project Members and Software Developers)*
- *Describe the overall scope of the report (e.g. detailed design decisions can be delegated to D4.2 Design Report)*

1.2 Implications from WP1

- *Describe the implications of D1.2 and D1.3 upon the PuppyIR Framework specification*
- *Discuss the software engineering challenges arising from WP1 (D1.2 and D1.3)*
- *Explain the need for a flexible software framework solution*
- *Describe the nature of software frameworks and systems constructed from frameworks*
- *Discuss the challenges such a framework faces in satisfying the agreed requirements*

1.3 Specification Structure

The specification report is divided into three major sections:

- Section 2 describes the **system context** of the PuppyIR Framework. The system context view of PuppyIR analyses the framework as a complete and closed system, with the architectural components hidden from view. The users, roles and external systems of the framework are represented along with the typical information flows (high-level use cases) that exist between them.
- Section 3 describes the global set of **design considerations** and **constraints** that will govern the design and development of the PuppyIR Framework.
- Section 4 describes the **software architecture** of the PuppyIR Framework. This section extends from the system context provided in Sec. 2 by describing the architectural strategy adopted to deliver a software framework that meets the project requirements. As the software architecture for any medium to large system cannot be satisfactorily conveyed by a single diagram or description style, we present three complementary views of the software architecture for PuppyIR: architectural overview that identifies the main architectural layers; a functional architecture that reveals the major sub-systems and components required by the framework; and a physical architecture that maps the components of the framework to physical devices and networks.

2 System Overview of PuppyIR

The first design artefact to consider is the system context of the PuppyIR Framework. This high-level view treats the PuppyIR Framework as a *black box system* and documents the interactions with external entities, such as users, access channels and other external systems. It also illustrates the information and control flows between the system and the external entities. This view of the system forms a link between the business context of what a service should resemble from a stakeholder perspective, to the more detailed and technical functional and operational views of the system architecture that focus on the internals of the PuppyIR Framework.

2.1 System Context

The system context of the PuppyIR framework is shown in the diagram below:

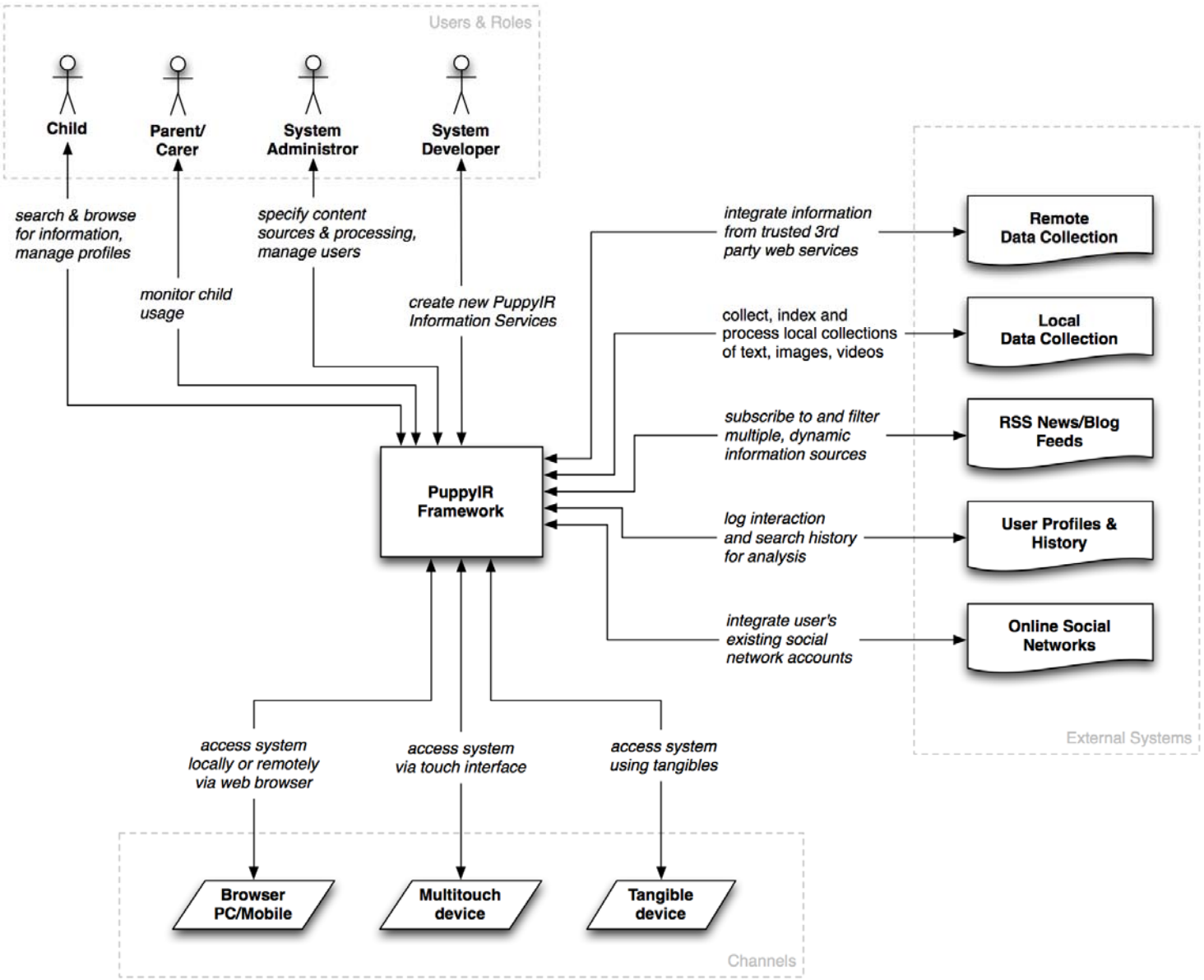

Figure 1: The System Context of PuppyIR

In this view of PuppyIR, the framework is considered as a black box, hiding the internal architecture and components of the framework, and only considering the users and roles that access PuppyIR, and in which contexts they can access it. Besides users and roles, the external systems that PuppyIR interacts with are also depicted. The following section describes each of these external entities in the context of PuppyIR.

2.2 Users and Roles

As PuppyIR is a framework, used to build information services, it necessarily has two distinct groups of users. The first grouping is children and their carer/parents. As described in **D1.2**, this group of users are the '**end-users**' of a built PuppyIR information service and have their own set of requirements. The second grouping is the **technical users**, the administrators of a system and the developers who will create the system from the components of the framework. Similarly, in **D1.3**, the technical requirements of this second group of users were reported.

2.2.1 Child

- *Description of role and context of access*
- *Description of information for which role can access*
- *Description of typical usage levels*

2.2.2 Carer/Parent

- *As above*

2.2.3 System Administrator

- *As above*

2.2.4 System Developer

- *As above*

2.3 Access Channels

The range of indicative scenarios and particular use cases of PuppyIR explored in **D1.2** suggests a wide range of potential access channels through which a user (primarily children) might interact with an information service. A central aim of PuppyIR is to explore novel forms of interaction modes that may aid children in satisfying their information needs, therefore it is crucial to support the flexibility to extend the range of access channels that can be integrated with PuppyIR.

2.3.1 Browser (or Kiosk) on PC/Mobile

- *Description of channel type and how the users make use of it*
- *Description of how the channel is integrated to the framework*
- *Description of any bandwidth needs or access protocols*

2.3.2 Multi-touch Devices

- *As above*

2.3.3 Tangible Devices

- *As above*

2.4 Interaction with External Entities

The external entities of the PuppyIR Framework mostly comprise of the sources of data that can be used as part of an information service. Although these are treated as external, this does not suggest that they all lie outside the system-boundary of a PuppyIR service – clearly information on users on a local database must be carefully protected as with any sensitive data, however this is considered to be external to the actual framework itself.

2.4.1 Remote Data Collections

- Description of the external system
- Description of 'placement' of external system in relation to PuppyIR (e.g. local/remote)
- Description of interaction with PuppyIR
- Description of any access protocols for interaction
- Description of any data formats or conversion matters
- Description of any compliance requirements (fair usage, legal, licensing, etc)
- Description of any non-functional concerns (security, availability, throughput, etc)

2.4.2 Local Data Collections

- As above

2.4.3 RSS (News/Blogs/etc) Feeds

- As above

2.4.4 User Profiles and Histories

- As above

2.4.5 Online Social Networks

- As above

2.5 Information and Control Flow

The previous sections have captured the static view of the overall system context, by describing the users and roles, access channels and external systems. This section describes the relationships between these entities and the PuppyIR framework (shown as labelled arrows on **Fig. 1**) as samples of 'information and control flow' that are present in the system context. Rather than give the precise sequence of interactions involved, each information flow will help illustrate how information flows between the system and the users, channels and external systems.

(*May represent as high level use-cases*)

2.5.1 User-oriented Information Flow

This section describes the information and control flow from the perspective of the users

2.5.1.1 Search and Browse for Information

- Description of the information that is flowing between the relevant entities during the transaction
- Categorise information flow as batch, interactive or real-time
- Description of type of data involved
- Estimate of volume of data in transaction

- Estimate of frequency transaction

2.5.1.2 Manage Profile and History

- As above

2.5.1.3 Monitor Child Usage

- As above

2.5.1.4 Specify Information Sources

- As above

2.5.1.5 Specify Information Processing

- As above

2.5.1.6 Manage Users

- As above

2.5.1.7 Create New PuppyIR Information Service

- As above

2.5.2 Channel-oriented Information Flow

This section describes the information and control flow from the perspective of the access channels.

2.5.2.1 Access system locally or remotely via web browser (or Kiosk)

- Description of the information that is flowing between the relevant entities during the transaction
- Categorise information flow as batch, interactive or real-time
- Description of type of data involved
- Estimate of volume of data in transaction
- Estimate of frequency transaction

2.5.2.2 Access system via touch interface

- As above

2.5.2.3 Access system using tangibles

- As above

2.5.3 External Systems-oriented Information Flow

This section describes the information and control flow from the perspective of the external systems.

2.5.3.1 Integrate information from trusted 3rd party web services

- Description of the information that is flowing between the relevant entities during the transaction
- Categorise information flow as batch, interactive or real-time
- Description of type of data involved
- Estimate of volume of data in transaction

2.5.3.2 Estimate of frequency transaction

- As above

2.5.3.3 Collect, index and process local collections of text, images, videos

- As above

2.5.3.4 Subscribe to and filter multiple, dynamic information sources

- As above

2.5.3.5 Log interaction and search history for analysis

- As above

2.5.3.6 Integrate user's existing social network accounts

- As above

3 Design Considerations

The set of general design considerations that will govern the development of the PuppyIR Framework are as follows:

3.1 Assumptions and Dependencies

- Related software / hardware
- Operating Systems
- Programming Languages
- Adopted Standards

3.2 General Constraints

- End-user environment (e.g. museums, home, hospitals and schools)
- Hardware / Software environment (e.g. device specifications, hosting services)
- Network communications (e.g. required bandwidth to
- Interface/protocol requirements

3.3 Goals and Guidelines

- Emphasise usability and user experience
- Simple is better (KISS-principle)
- Reuse relevant components
- Minimise redundancy (DRY-principle)
- Code clearly and document comprehensively

3.4 Development methodology

- Rational Unified Process
- Agile Methodology
- Small development team
- Use SCM

4 System Architecture

The previous sections have considered the system context and design considerations for the PuppyIR Framework. This section focuses on the architecture that will guide the design and development of the framework. To capture the software architecture of PuppyIR, three complementary high-level views are explored:

- Architecture Overview – illustrate the main conceptual elements of PuppyIR
- Functional Architecture – illustrates how the components of PuppyIR interact
- Operational Architecture – illustrates how PuppyIR is mapped to physical systems

The following sections describe these three architectural views of PuppyIR.

4.1 Architecture Overview

The layered architecture **[ref]** of the PuppyIR Framework is shown in the diagram below.

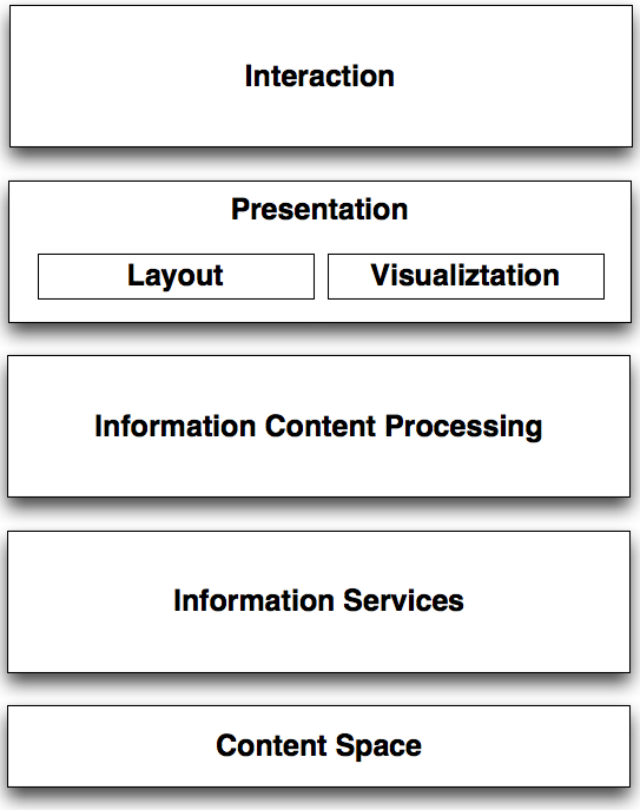

Figure 2: Layered Architecture of PuppyIR

4.1.1 Interaction Layer

The Interaction Layer is concerned with components that support the variety of modes and modalities that a user may utilise when using a PuppyIR service, ranging from familiar textual and graphical interfaces, onto more novel multi-touch and tangible interfaces.

- Description of concerns of this layer
- Description of relationship to other layers

4.1.2 Presentation Layer

The Presentation Layer is concerned with the components that enable the novel presentation of the search interfaces (e.g. for querying, viewing results and browsing/exploring) for a PuppyIR service.

- Description of concerns of this layer
- Description of relationship to other layers

4.1.3 Information Processing Layer

The Information Processing Layer is concerned with the components that are responsible for intelligent processing of the available information (user queries, user profiles, user histories, local data, remote data and dynamic feeds) within a PuppyIR service. Typical processing components include summarisation, rating, filtering, mining, and clustering.

- Description of concerns of this layer
- Description of relationship to other layers

4.1.4 Information Services

As there are a variety of information services that PuppyIR can use, the Information Services Layer provides service adapters and data-format interchange components to integrate multiple sources of information if required. Information Services may be either locally or remotely accessible. A local service could be a search engine (e.g. Lucene or Terrier etc) that has indexed a large collection of data and is to be integrated into a PuppyIR service. Alternatively, a remote service could be a web service that supports search requests and returns results to the client.

- Description of concerns of this layer
- Description of relationship to other layers

4.1.5 Content Space

The Content Space Layer is concerned with the collections of information that are used by a PuppyIR service and the translation of data-formats to a common data model. The content may be large repositories of relatively static remote data, local data, or data from dynamic sources such as RSS feeds generated by blog and news sites, which is generally much small but more rapidly updated.

- Description of concerns of this layer
- Description of relationship to other layers

4.2 Functional Architecture

The functional architecture of the sub-systems and components that comprise the PuppyIR Framework are shown in the diagram below.

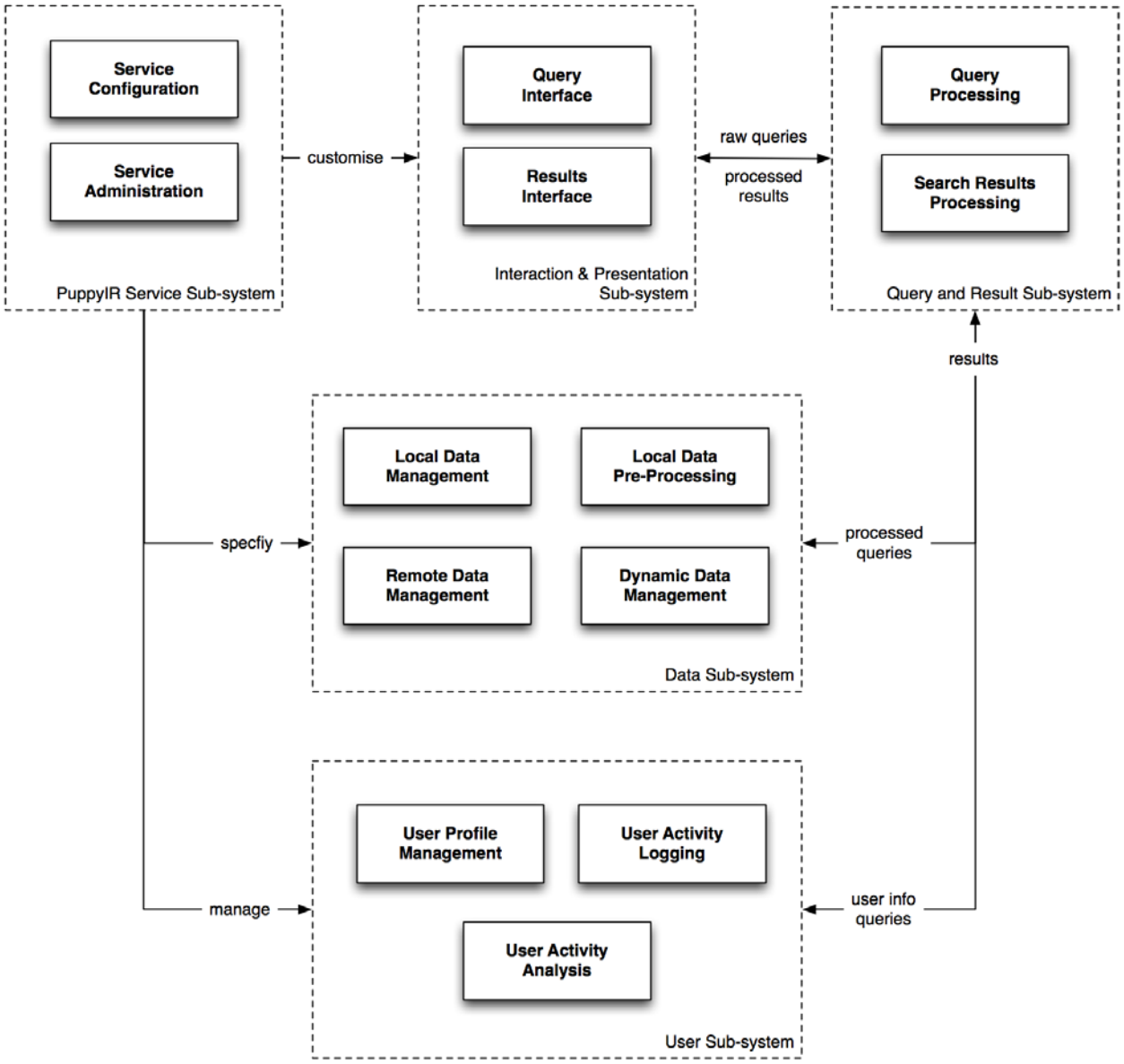

Figure 3: Functional Architecture of PuppyIR Framework

4.2.1 PuppyIR Service Sub-system

- Introduce and describe the sub-system
- Describe the functional components

4.2.1.1 Service Configuration

- Component description
- Component responsibilities
- Etc.

4.2.1.2 Service Administration

4.2.2 Interaction and Presentation Sub-system

- Introduce and describe the sub-system
- Describe the functional components

4.2.2.1 Query Interface

4.2.2.2 Results Interface

4.2.3 Query and Result Sub-system

- Introduce and describe the sub-system
- Describe the functional components

4.2.3.1 Query Processing

4.2.3.2 Search Result Processing

4.2.4 Data Sub-system

- Introduce and describe the sub-system
- Describe the functional components

4.2.4.1 Local Data Management

4.2.4.2 Local Data Pre-processing

4.2.4.3 Remote Data Management

4.2.4.4 Dynamic Data Management

4.2.5 User Sub-system

- Introduce and describe the sub-system
- Describe the functional components

4.2.5.1 User Profile Management

4.2.5.2 User Activity Logging

4.2.5.3 User Activity Analysis

4.3 Operational Architecture

The operational architecture of the PuppyIR Framework (as viewed as a completed information service) is shown in the following diagram.

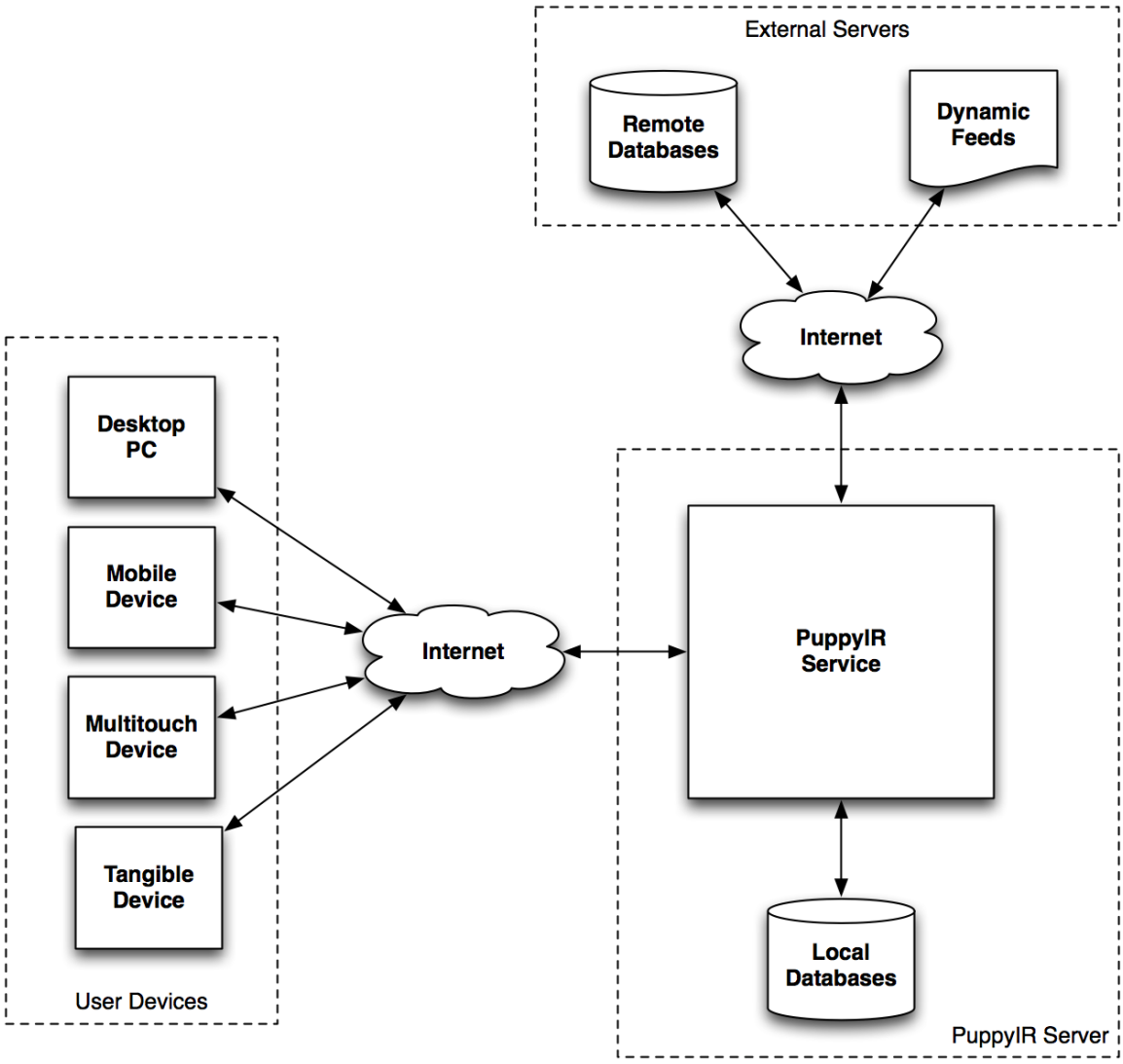

Figure 4: Operational Architecture of the PuppyIR Framework

4.3.1 PuppyIR Server

- Describe the class of physical device
- Describe how the components of the Framework relate to this device

- Describe the expected connectivity to other devices in the system

4.3.2 External Servers

- As above

4.3.3 User Devices

- As above

5 Policies and Tactics

This section covers the non-architectural decisions that can be made without affecting the overall design, but are important to be documented for the PuppyIR Framework.