The following sections describe the key concepts associated with Troux products and provide information about the product architecture.

Overview .................................................................................................................................. 2
TrouxSource Repository .................................................................................................... 2
Troux Semantics Metamodel ............................................................................................. 2
Troux Navigate .................................................................................................................. 4
Troux Insight ..................................................................................................................... 4
Troux Architect .................................................................................................................. 4
Data Integration .................................................................................................................. 5
Product Architecture ................................................................................................................. 6
Client Tier ........................................................................................................................... 6
Application Tier .................................................................................................................. 7
Database Tier ....................................................................................................................... 7
Communications and Data Paths ............................................................................................. 8
Fault Tolerance and Redundancy ............................................................................................ 9
Troux Collection Data Flow .................................................................................................... 10
Administrative Roles within Troux .......................................................................................... 11
Third-Party Technologies ....................................................................................................... 11
Open Source and Public Domain Technologies ...................................................................... 12
Licensed Technologies ......................................................................................................... 12
Overview

Troux provides an integrated suite of products used to view, analyze, and manage enterprise-level portfolios. The following sections provide an overview of the key components within the Troux product suite.

TrouxSource Repository

The TrouxSource repository is an enterprise-scale repository for storing large quantities of data. The TrouxSource repository provides a central repository for storing both of the following types of data:

- **Metamodel data**: Troux Semantics type definitions. For more information, see Troux Semantics Metamodel below.
- **Model data**: The components and relationships that represent an enterprise’s assets.

Model and metamodel data in the TrouxSource repository can be accessed through multiple client interfaces (described in the following sections). Depending on the user’s business needs, these clients can be used alone or in conjunction.

Troux Semantics Metamodel

A *metamodel* is a framework that defines a set of modeling elements and specifies how those elements interrelate. The metamodel provides a library of component and relationship types. These types determine what elements can be represented in models based on that metamodel.

- Component types represent physical and conceptual entities such as hardware, applications, projects, and business capabilities. Each component type has a set of properties specific to that type. For example, a project has a *Phase* property while a person has a *Gender* property.

- Relationship types define which component types can be connected and the nature of that connection, as shown in Figure 2. Relationship types can also have properties.
The Troux Semantics metamodel enables Troux users to represent extensive enterprise portfolios and to model complex environments. Troux Semantics defines a wide variety of types based on the best practices of Global 2000 enterprises and government agencies. If necessary, users can extend the metamodel to include custom types or properties that meet their organization’s specific needs.

The Troux Semantics metamodel is subdivided into smaller metamodels, called domains. The domains can be categorized as asset domains (those associated with physical assets) and concept domains (those associated with intellectual property, processes, or skills), as shown in Table 1.

<table>
<thead>
<tr>
<th>Asset Domains</th>
<th>Concept Domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application and Software Domain</td>
<td>Analysis Domain</td>
</tr>
<tr>
<td>Data Domain</td>
<td>Financial Domain</td>
</tr>
<tr>
<td>Document Domain</td>
<td>General Domain</td>
</tr>
<tr>
<td>Governance Domain</td>
<td>Information Domain</td>
</tr>
<tr>
<td>Infrastructure and Hardware Domain</td>
<td>IT Architecture Domain</td>
</tr>
<tr>
<td>IT Product Domain</td>
<td>IT Patterns Domain</td>
</tr>
<tr>
<td>IT Service Domain</td>
<td>Knowledge and Skill Domain</td>
</tr>
<tr>
<td>Location Domain</td>
<td>Market Domain</td>
</tr>
<tr>
<td>Organization Domain</td>
<td>Policy Domain</td>
</tr>
<tr>
<td>Product and Service Domain</td>
<td>Process Domain</td>
</tr>
<tr>
<td>Service Architecture Domain</td>
<td>Resource Domain</td>
</tr>
<tr>
<td>Services Portfolio Management Domain</td>
<td>Strategy Domain</td>
</tr>
<tr>
<td>Systems Domain</td>
<td>Timeline Domain</td>
</tr>
<tr>
<td>Troux Archetype Domain</td>
<td>Transition Domain</td>
</tr>
<tr>
<td>Troux Information Domain</td>
<td></td>
</tr>
</tbody>
</table>

Because Troux Semantics provides a “common language” with which to describe enterprise assets, model data can be created, viewed, or modified through multiple interfaces.
Troux Navigate

Troux Navigate is a server-based web application with a configurable, browser-based interface. Troux Navigate enables users to:

- **Access Troux Portfolios.** Troux Portfolios is automatically installed with Troux Navigate. It allows users to browse, search, edit, and report against model data stored in the TrouxSource repository. Troux Portfolios provides a set of targeted data management capabilities and decision support tools centered around specific portfolios and business goals. For example, Troux Portfolios provides a set of reports, data management interfaces, and predefined workflows designed to help users optimize their application portfolios.

- **Access other solutions.** In addition to Troux Portfolios, organizations can install other Troux solutions, or they can configure custom solutions to address their specific needs. These solutions are accessed through Troux Navigate.

- **Perform administrative tasks,** such as configuring custom solutions, defining policy rules to govern data management, and submitting data collection jobs.

Troux Insight

Troux Insight is a server-based web application accessed through a browser-based interface. Troux Insight provides powerful ad-hoc analysis capabilities of the model data stored in the TrouxSource repository. By clearly showing connections between components from different portfolios, Troux Insight helps users eliminate “information silos” and ensure that decisions are aligned with business priorities.

Troux Architect

Troux Architect is native Windows application used to create visual models (see Figure 3). Visual representations of model data can help users quickly understand connections and dependencies between different types of elements.

Because they are based on Troux Semantics, the components and relationships in a Troux Architect model store property values and adhere to the connection rules defined by the metamodel. Integration with the TrouxSource repository enables Troux Architect users to interact with model data stored in the repository. Users can create new objects and relationships in Troux Architect and commit them to the TrouxSource repository. Additionally, users can search the TrouxSource repository for existing components and pull these components into a Troux Architect model. Any changes made to components in Troux Architect can be committed back to the repository.
Data Integration

Troux provides the following methods for integrating large quantities of data:

- **Data Collection** provides tools for collecting data from a variety of sources for import into the TrouxSource repository. Data collection jobs, which are managed on the Troux server, aggregate and upload data to support enterprise-wide views and analytics.

- **TUX export mechanisms** and specialized datamart exports can be used to export data from the TrouxSource repository to external sources.

**NOTE**

Metamodel information is stored in both the TrouxSource repository and Troux Architect. Any metamodel extensions are defined in Troux Architect and imported into the TrouxSource repository.
Product Architecture

The Troux product suite is implemented in a three-tier architecture, as shown in the following diagram. Each tier is discussed below.

Client Tier

The client tier includes the following interfaces:

- **Browsers**: Troux applications with browser-based interfaces can be accessed through any supported Web browser.

- **Troux Architect**: A native Windows application used to create visual models. (See Troux Architect on page 4.)

![Troux Product Suite’s Three-Tier Architecture](image)

Figure 4: Troux Product Suite’s Three-Tier Architecture

Client Tier
- **BPClient**: A remote programmatic agent used to upload data.

### Application Tier

#### Application Server

The application tier includes Troux’s application server, which is a preconfigured Tomcat application server. The following applications run on the application server:

- **TIP**: which enables access to the Troux Navigate and Troux Insight client interfaces. TIP also hosts Troux Portfolios and possibly other Troux solutions, which are accessed through Troux Navigate. TIP also provides database caching.

- **BPServer**: which supports administrative functions such as importing and exporting system data.

- **bpstatic**: which serves up static content such as images and scripts.

The application server provides the following services to the applications that run in it:

- Database connection pooling.

- Authorization. The application server integrates with your corporate authentication facility. Authentication is done externally, while authorization is handled internally. When a user has been authenticated, the user name is passed to the application server. The Troux User Factory determines which Troux Groups the user belongs to and creates the user session.

#### TrouxAdmin

TrouxAdmin is a command line interface for performing various administration tasks, such as updating the metamodel and running data collection.

### Database Tier

The database tier is contained within an RDBMS. It includes the following elements:

- **TrouxSource Repository Database**: A central storage facility for TrouxSource repository data, including type definitions, model data, authorization information (role definitions and assignments), application configurations, and queries.

NOTE

The metamodel is stored in the database at a level of abstraction above the model data. Changes to component and relationship type definitions in the TrouxSource repository do not require changes to RDBMS schema.

Troux supports both Oracle and SQL Server deployments.

- **Troux Datamart**: A preconfigured datamart that supports Troux reports.

- **Custom Datamarts**: User-defined datamarts that support custom reports. Users can choose to deploy one or more custom datamarts as needed.
Communications and Data Paths

Troux components use the following communications protocols to pass data:

- Communications between the **client tier** and the **application tier** go over HTTP or HTTPS connections.
  - TIP sends and receives user and administrative data (HTML and XML).
  - BPServer sends and receives administrative data (XML).
  - Collector clients upload TUX files (XML) in the course of data collection.
  - Troux Architect sends and receives metamodel and model data (XML).

- Communications between the **application tier** and the **database tier** use JDBC over TCP connections.
Firewall Considerations

Troux does not have specific firewall requirements. Firewall deployment should be based on your organization’s policies and the needs of your environment.

Fault Tolerance and Redundancy

Load balancing techniques can be used to provide multiple redundant paths. The standard license agreement for Troux supports installation on multiple servers to facilitate deployments that require fault tolerance and redundancy.

Figure 7 illustrates a large production environment that is configured for maximum fault tolerance. In addition to clustering and load balancing it provides for additional fault tolerance through two mechanisms:

- IP failover for the servers
- Transactional replication for the database

These methodologies support geographic failover. If one facility fails (for example, from sustained power outage or natural disaster) the production system can continue to run from the failover facility.
Troux Collection Data Flow

Troux Collection allows you to automate data collection from external systems and import it into the TrouxSource repository. Data collection is accomplished using Troux’s Universal Collector Framework (UCF) and Troux’s data load facility, called TUX. TUX is an ETL (extract, transform, and load) facility designed to take upload files in XML format as input. From these files, TUX creates or updates components and relationships in the TrouxSource repository.

**NOTE**
For more information, see the *Data Collection Guide*.

The Troux Collection data flow (shown in Figure 8) includes the following steps:

1. The data source is accessed via the appropriate means (JDBC, ODBC, API, etc.).
2. UCF generates a TUX file.
3. The TUX file is processed within the Troux Server. Data is first staged into the database to better facilitate bulk processing. The Transform stage includes processing to locate preexisting components. During the Load stage, components and relationships are created or updated.
Administrative Roles within Troux

In most deployments, two types of administrators are responsible for administering the Troux system:

- The systems administrator is responsible for deployment, and for ensuring system availability.
- The Troux administrator is responsible for authorization administration, or assigning users to Troux groups and roles.

Third-Party Technologies

Troux software includes a variety of technologies developed by third parties. Key technologies are listed in the following sections. For a complete listing, see the About box in Troux Navigate.
Open Source and Public Domain Technologies

- Apache technologies
  - Tomcat (servlet container)
  - ANT (scripting)
  - POI (access to Microsoft files)
  - Struts (web application framework)
- Java Software Development Kit, Standard Edition
- Business Intelligence Reporting Tool (BIRT), which is available under the terms of the Eclipse Public License published by the Eclipse Foundation (www.eclipse.org)

Licensed Technologies

- SSL Encryption support is provided using code from RSA Security, Inc.
- SQL Server JDBC driver
- Oracle JDBC driver