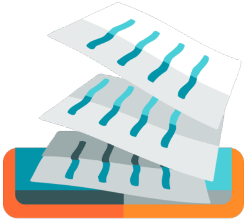
Summary: The data storage structure of nodes within the network hierarchy controls where and how

the bots you run store the data they produce.

Foundations->Node->Data Storage->Definition



The data storage hierarchy holds references to all the data bots have produced.

Foundations->Node->Data Storage->Content

Think of this hierarchy like a treasure chest. Inside this chest is kept all the valuable datasets generated within this workspace in one convenient place so that it can be used across the system by other bots, trading systems, etc.

The main types of data held under this hierarchy are:

- Learning Mines Data
- Trading Mines Data
- Data Mines Data
- Portfolio Mines Data

As you look down the hierarchy you will see that data is organized according Mine Type, Project, Exchange, and Market. This helps to keep the vast array of data used within Superalgos accessibly and understandable. The nodes under this hierarchy that hold specific sets of data are:

-

- Bot Products
- Trading Session Reference
- Learning Session Reference
- Portfolio Session Reference

Data Mines Data Section

Summary: The Data Mines Data section of the hierarchy controls where and how the sensor and indicator bots store the data they produce.

Foundations->Node->Data Mines Data->Definition

Session independent data refers to data generated by sensors and indicators, not related to trading sessions.

Foundations->Node->Data Mines Data->Content

As explained in the sorting of tasks page, data mines data is sorted by exchange, market, and the corresponding data mine, sensor or indicator bot, and bot product. That is, most of the nodes in this section of the hierarchy play an organizational role.

Many of them require references to the nodes that delimit the context for which the data is applicable. For example, the market data products node must reference one of the installed markets, in particular, the market on which the data mining operation is run. These references help other entities understand the context to which the data belongs to.

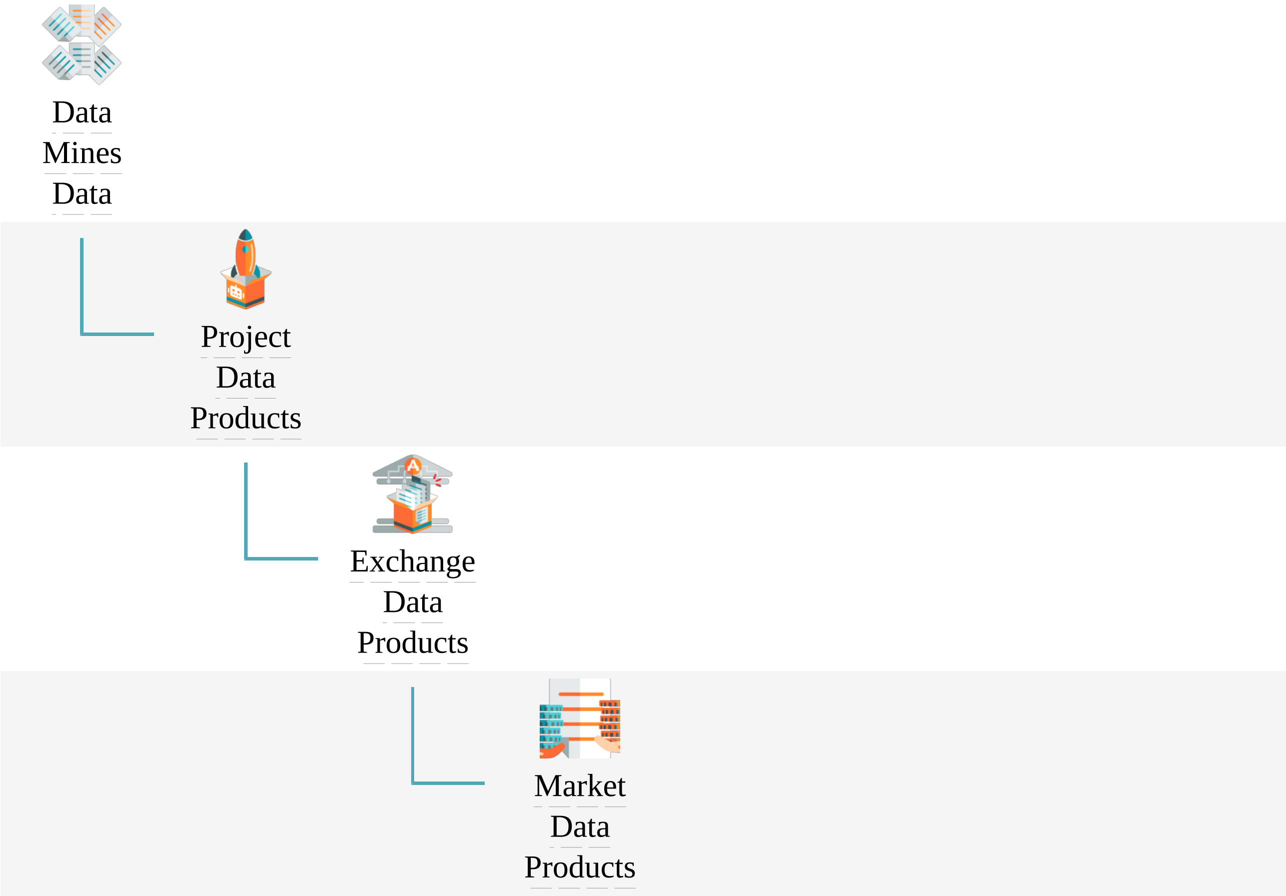
Whenever you create data mining tasks manually from within the Network hierarchy (as opposed to using the install market function under the exchange markets node of the

Crypto Ecosystem

hierarchy

Foundations->Concept->Reusable Snippets->Note for Hierarchy Tables

Note: This page discusses the top-level information about the section of the hierarchy represented in the below table. Click on any of the nodes to get the details, including particulars on their configuration, when applicable.



Data Mines Data

Project Data Products

Exchange Data Products

Market Data Products

Project Data Products

Foundations->Node->Project Data Products->Definition

The

Project Data Products

node is an organizational device grouping all

Bot Products

corresponding to the same

Project

.

Exchange Data Products

Foundations->Node->Exchange Data Products->Definition

The exchange data products node organizes data mines data by exchange. That is, each exchange installed in the system has an exchange data products node grouping all market data products nodes corresponding to the said exchange.

Foundations->Node->Exchange Data Products->Content

The exchange data products node must reference the exchange of choice. This reference constraints the rest of the definitions to the context of the said exchange.

When representing an exchange featured in the system's icons library, the standard exchange data tasks icon is replaced by the exchange's logo.

Market Data Products

Foundations->Node->Market Data Products->Definition

A market data products node represents the group of data products generated in a specific market.

Foundations->Node->Market Data Products->Content

The market data products node must reference a market defined in the

Crypto Ecosystem

hierarchy.

This node may spawn individual data products or may deploy data products in bulk organized by data mine and by bots. See the data products node for the details.

Trading Mines Data Section

Summary: The

Trading Mines Data

section of the hierarchy controls where and how the

Trading Bot

stores the data produced.

Foundations->Node->Trading Mines Data->Definition

Trading Mines Data

refers to data that is generated as a consequence of running a trading session, that is, data the trading bot instance generates while running backtesting, paper trading, forward testing, or live trading sessions.

Foundations->Node->Trading Mines Data->Content

Trading

mines data is sorted by exchange, market, the corresponding trading session, and the corresponding trading mine, trading bot, and bot product. That is, most of the nodes in this section of the hierarchy play an organizational role.

Many of them require references to the nodes that delimit the context for which the data is applicable. For example, the exchange trading products node must reference one of the installed exchanges, in particular, the exchange on which the trading operation is run. These references help other entities understand the context to which the data belongs to.

Whenever you create trading tasks manually from within the Network hierarchy (as opposed to using the install market function under the exchange markets node of the

Crypto Ecosystem

hierarchy

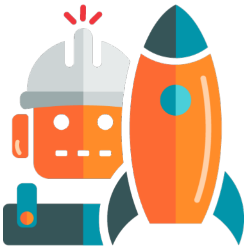
Foundations->Concept->Reusable Snippets->Note for Hierarchy Tables

Note: This page discusses the top-level information about the section of the hierarchy represented in the below table. Click on any of the nodes to get the details, including particulars on their configuration, when applicable.

- Trading Mines Data
    - Project Trading Products
        - Exchange Trading Products
            - Market Trading Products
                - Trading Session Reference

Project Trading Products

Foundations->Node->Project Trading Products->Definition



The

Project Trading Products

node is an organizational device grouping all

Bot Products

corresponding to the same project.

Exchange Trading Products

Foundations->Node->Exchange Trading Products->Definition

The exchange trading products node organizes trading mines data by exchange. That is, each exchange installed in the system has an exchange trading products node grouping all session references corresponding to the said exchange.

Foundations->Node->Exchange Trading Products->Content

The exchange trading products node must reference the exchange of choice. This reference constraints the rest of the definitions to the context of the said exchange.

When representing an exchange featured in the system's icons library, the standard exchange trading products icon is replaced by the exchange's logo.

Market Trading Products

Foundations->Node->Market Trading Products->Definition

A market trading products node features the group of data products generated by the referenced session in a specific market.

Foundations->Node->Market Trading Products->Content

The market trading products node must reference a market defined in the

Crypto Ecosystem

hierarchy.

This node may spawn individual data products or may deploy data products in bulk organized by trading mine and by trading bots. See the data products node for the details.

Trading Session Reference

Foundations->Node->Trading Session Reference->Definition

A trading session reference establishes which instance of a trading system will be the one that stores data in the current location.

Foundations->Node->Trading Session Reference->Content

A such, the trading session reference node must establish a reference with a trading session. Also, its offspring nodes determine precisely which data products are stored.

Data and Trading Mine Products

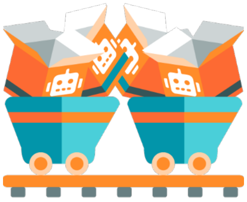
Summary: Both the

Data Mine Products

and

Trading Mine Products sections under Data Mines Data and Trading Mines Data respectively have a very similar arrangement, and store data in the same type of nodes. We will review both sections side by side.

Foundations->Node->Data Mine Products->Definition



Data mine products are references established with data mines to facilitate establishing data product references with multiple products in the given data mine.

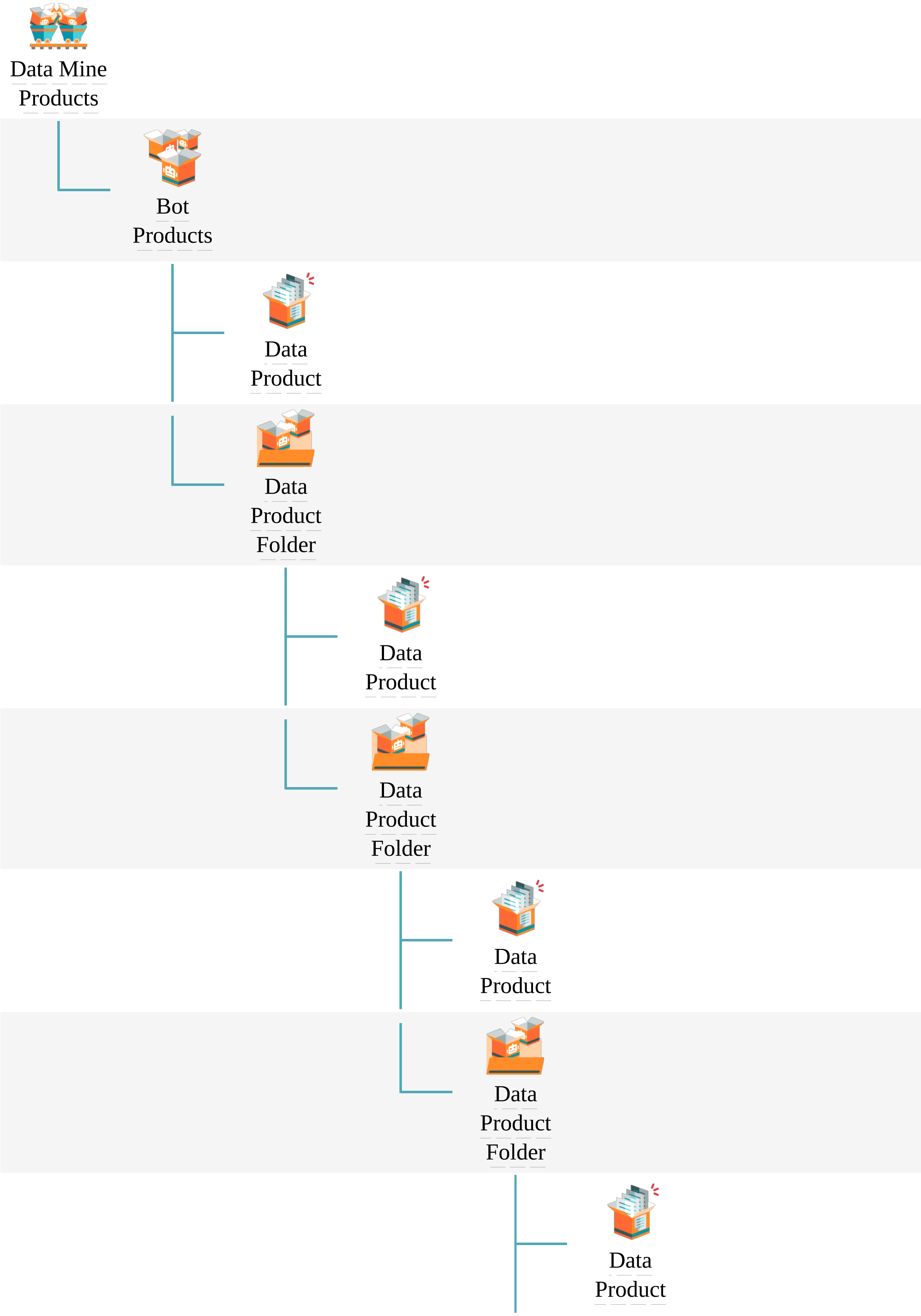
Foundations->Node->Data Mine Products->Content

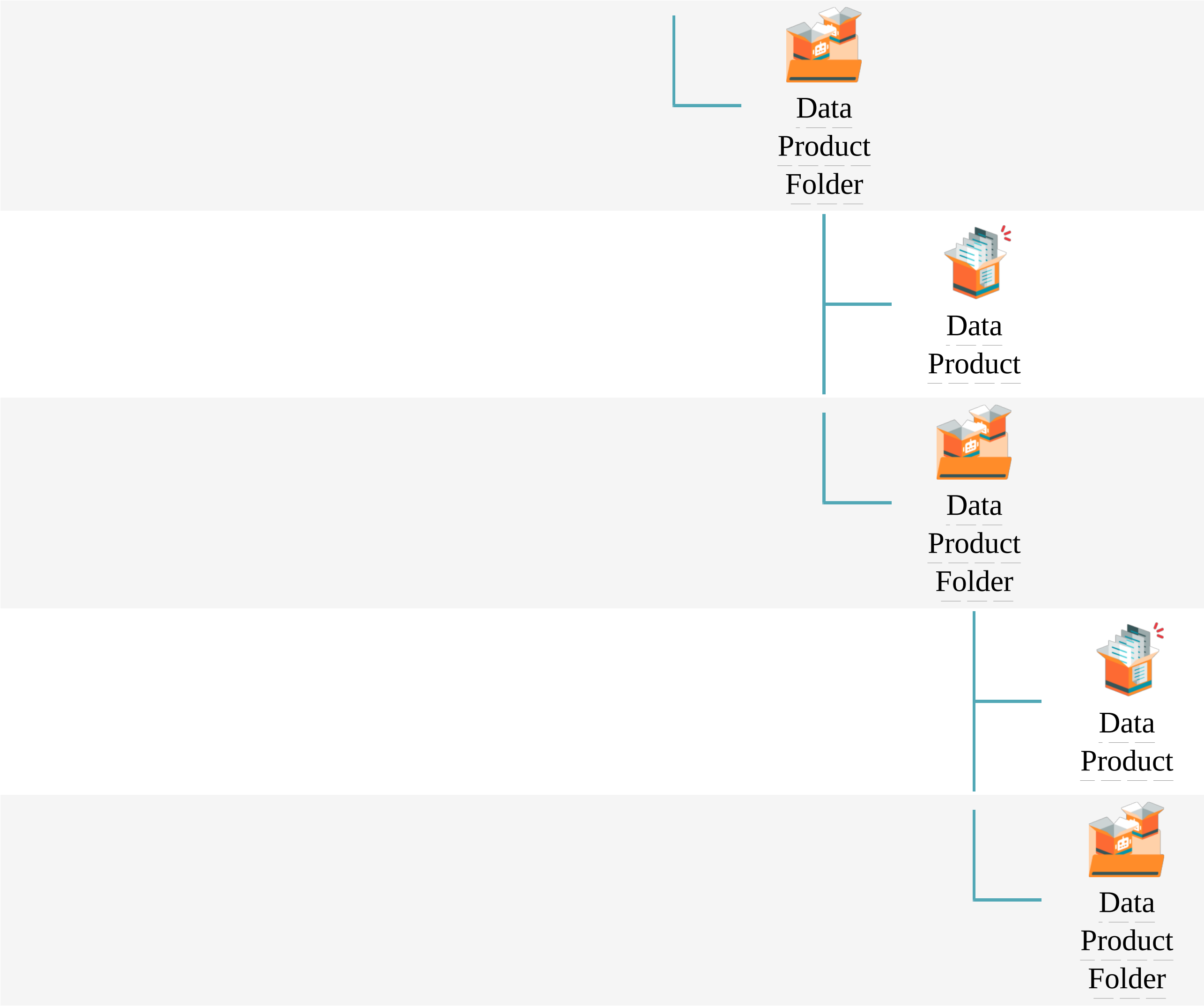
The node may be used as an organizational device, simply to arrange bot products. However, the smart use of the node involves automating the deployment of multiple data products.

The use of the data mine products node is optional, as data products may also exist outside of data mine products nodes.

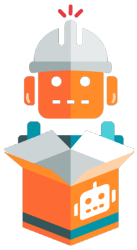
Foundations->Concept->Reusable Snippets->Note for Hierarchy Tables

Note: This page discusses the top-level information about the section of the hierarchy represented in the below table. Click on any of the nodes to get the details, including particulars on their configuration, when applicable.


Data Mine Products
Bot Products
Data Product
Data Product Folder
Data Product
Data Product Folder
Data Product
Data Product Folder
Data Product



Foundations->Node->Trading Mine Products->Definition



Trading

mine products are references established with trading mines to facilitate establishing data product references with multiple products in the given mine.

Foundations->Node->Trading Mine Products->Content

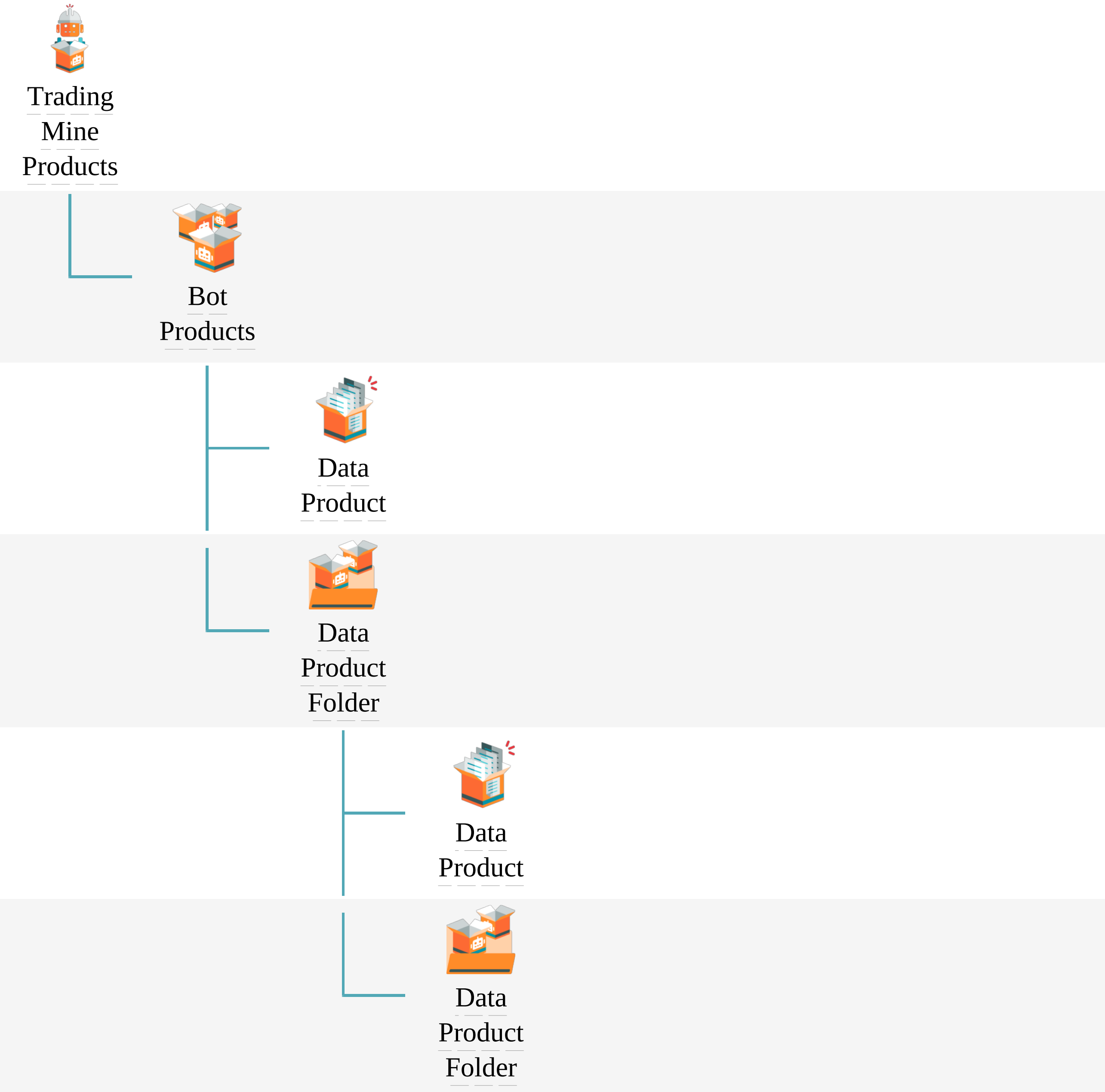
At this point, Superalgos ships with a single trading mine, featuring a single low frequency trading bot. However, developers may create their own trading bots or fork the existing one.

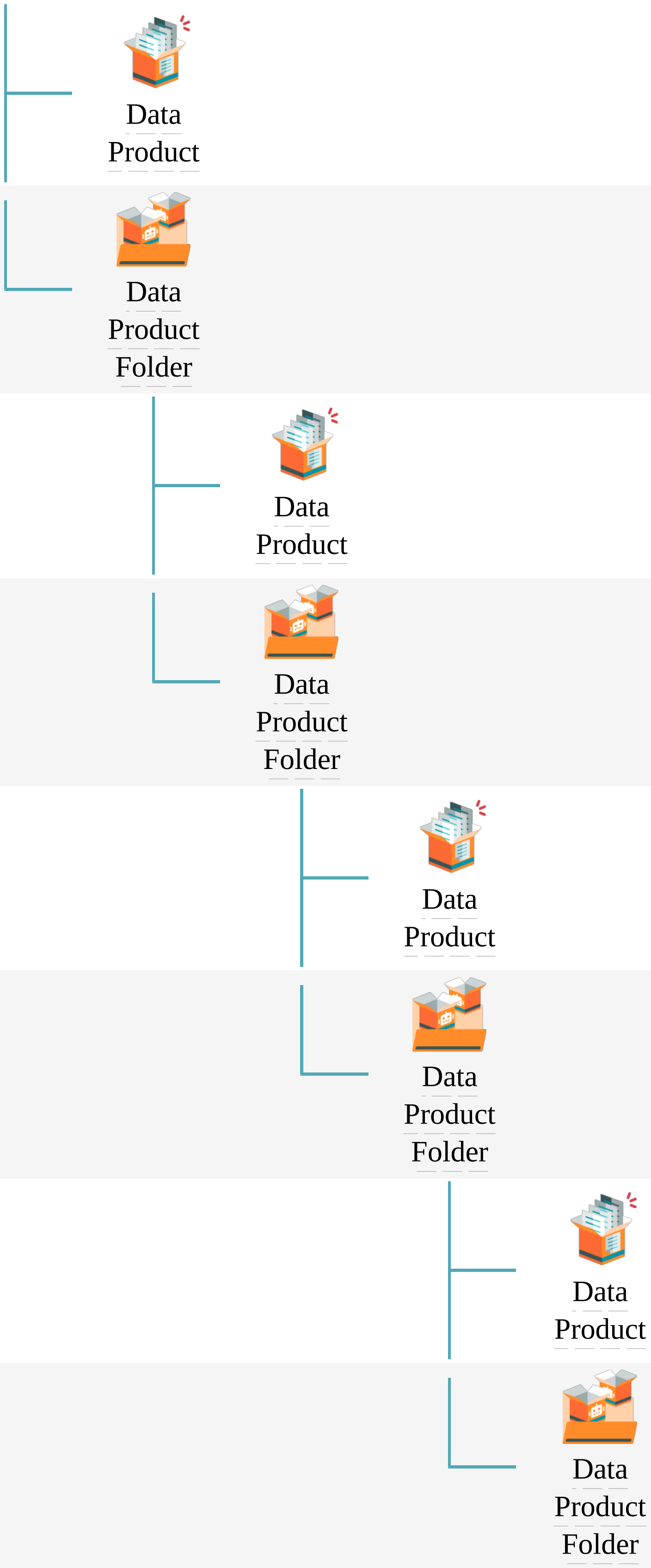
The node may be used as an organizational device, simply to arrange bot products. However, the smart use of the node involves automating the deployment of multiple data products.

The use of the trading mine products node is optional, as data products may also exist outside of trading mine products nodes.

Foundations->Concept->Reusable Snippets->Note for Hierarchy Tables

Note: This page discusses the top-level information about the section of the hierarchy represented in the below table. Click on any of the nodes to get the details, including particulars on their configuration, when applicable.





Bot Products

Foundations->Node->Bot Products->Definition

Data Product

Foundations->Node->Data Product->Definition

A data product represents the collection of datasets generated by the instance of a bot as defined in the corresponding data mine or trading mine.

Foundations->Node->Data Product->Content

Data products exist in the context of trading mines data and data mines data nodes. In the first case, a data product is the collection of datasets generated by an instance of a trading bot as defined in a trading mine, running a trading session. In the latter case, it is the collection of datasets generated by either a sensor bot or an indicator bot instance, as defined in a data mine, and running a data mining task.

A data product node must reference a product definition in the corresponding bot.

Important: Data products may only be stored on the same network node on which the task—data mining or trading—runs.

How to Safely Close Superalgos

Summary: The correct way to close Superalgos is to stop all tasks in the data-mining and trading operations, wait a few seconds until the tasks stop, and only then close the browser. Before closing the console, you must wait around a minute until there is no further activity.

The backend application runs independently from the frontend browser-based application. You may close the browser at any point without affecting your data-mining operation or the bots you may be running in your testing or production environments. Once you restart the browser and navigate to the correct URL, the frontend app reconnects with the backend and you regain control over the whole operation.

Important: If you wish to completely close Superalgos and stop all operations, there is a safe way to do this while preventing the corruption of files that may be processing at any point.

Start Here

1.

Close

all data-mining tasks. Select Stop All

Exchange Data Tasks

on the data mining node menu.

2.

Close

all tasks in the testing environment and the production environment. Select *Stop All*

Task

Managers in the testing and production environment nodes menus.

3. Wait a few seconds until all tasks are stopped and close the browser.

4. Wait for about one minute until all activity stops in the console, then close it.