Overview

This section contains the following topics:

- Cisco WAE Overview, on page 1
- WAE Architecture, on page 2
- WAE Applications, on page 5
- Cisco WAE Interfaces, on page 6
- Network Model Creation Workflow, on page 7

Cisco WAE Overview

The Cisco WAN Automation Engine (WAE) platform is an open, programmable framework that interconnects software modules, communicates with the network, and provides APIs to interface with external applications.

Cisco WAE provides the tools to create and maintain a model of the current network through the continual monitoring and analysis of the network and the traffic demands that are placed on it. This network model contains all relevant information about a network at a given time, including topology, configuration, and traffic information. You can use this information as a basis for analyzing the impact on the network due to changes in traffic demands, paths, node and link failures, network optimizations, or other changes.

The Cisco WAE platform has numerous use cases, including:

- Traffic engineering and network optimization—Compute TE LSP configurations to improve the network performance, or perform local or global optimization.
- Demand engineering—Examine the impact on network traffic flow of adding, removing, or modifying traffic demands on the network.
- Topology and predictive analysis—Observe the impact to network performance of changes in the network topology, which is driven either by design or by network failures.
- TE tunnel programming—Examine the impact of modifying tunnel parameters, such as the tunnel path and reserved bandwidth.
- Class of service (CoS)-aware bandwidth on demand—Examine existing network traffic and demands, and admit a set of service-class-specific demands between routers.

WAE Architecture

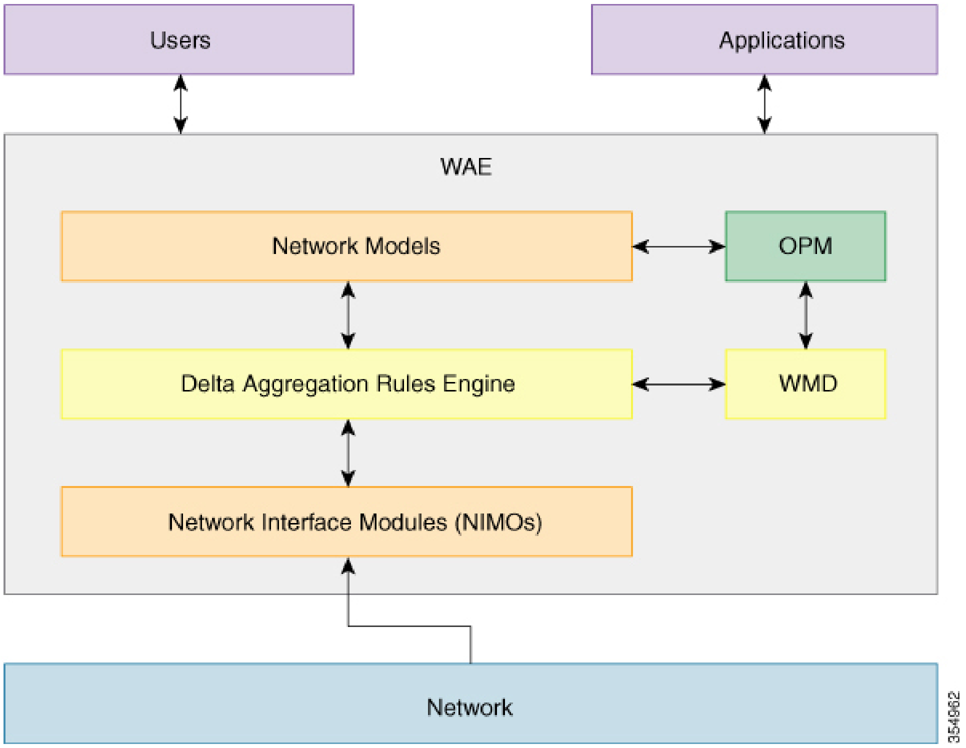
At its core, WAE defines an *abstract network model*, which can be built from an actual network by stitching together *network interface modules (NIMOs)*.

The WAE network model is defined in YANG and is *extensible* via standard YANG mechanisms. WAE itself is implemented on top of a YANG run-time system that automatically generates APIs (NETCONF, RESTConf, CLI) from the YANG models.

Optimization and prediction modules (OPMs) provide a powerful Python API to manipulate network models. The OPM API lets you operate on the network without having to worry about device-specific properties. Even if the underlying routers are replaced by routers from a different vendor, the API calls remain exactly the same.

The OPM APIs provide powerful "what-if" capabilities. For example, the OPM APIs let you answer the following questions:

- What is the impact if I bring this router down for maintenance?
- What happens if I increase the capacity of this circuit?
- Can my network handle a data center backup now?


Users
Applications
WAE
Network Models
OPM
Delta Aggregation Rules Engine
WMD
Network Interface Modules (NIMOs)
Network
354962

Optimization and Prediction Module

Optimization and prediction modules (OPMs) provides a powerful Python API to manipulate network models. The OPM API lets you operate on the network without having to worry about device-specific properties. Even if the underlying routers are replaced by routers from a different vendor, the API calls remain exactly the same.

The OPM APIs provides powerful ***what-if*** capabilities. For example, the OPM APIs let you answer the following questions:

- What is the impact if I bring this router down for maintenance?
- What happens if I increase the capacity of a particular circuit?
- Can my network handle a datacenter backup now?

Network Interface Modules

A *network interface module (NIMO)* is a WAE package that populates parts of the abstract network model, possibly querying the network to do so. Most NIMOs operate as follows:

1. They read a *source network model* (or simply, a *source model*).
2. They augment the source model with information obtained from the actual network.
3. They produce a *destination network model* (or simply, a *destination model*) with the resulting model.

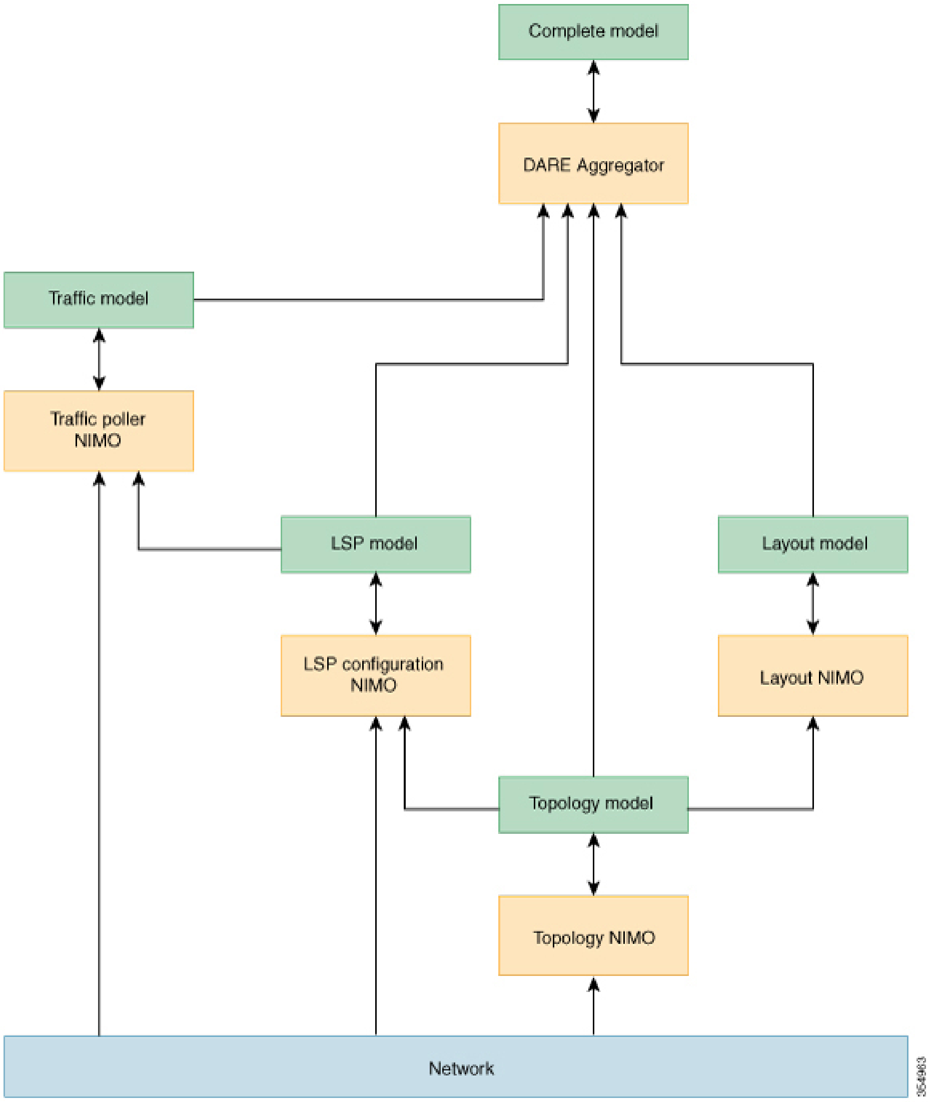
WAE includes several different NIMOs, such as:

- Topology NIMO—Populates a basic network model with topology information (nodes, interfaces, circuits) based on the discovered IGP database augmented by SNMP queries. The topology NIMO does not have a source model.
- LSP configuration NIMO—Augments a source model with LSP information, producing a destination model with the extra information.
- Traffic poller NIMO—Augments a source model with traffic statistics polled from the network, producing a new destination model with extra information.
- Layout NIMO—Adds layout properties to a source model to improve visualization. It produces a new destination model with the extra layout information. The NIMO records changes to the layout properties, so when the source model changes and the destination model is updated, the layout properties in the destination model are updated accordingly.

Network Models

A *model building chain* is an arrangement of NIMOs organized in such a way as to produce a network model with the desired information. Given the preceding NIMOs, for example, one chain could consist of the topology NIMO, followed by the LSP configuration NIMO, followed by the traffic poller NIMO. This chain contains three models, some with more information than others. Organizing model building chains lets you create different models for different use cases. You can branch the chain to have independent model building tracks.

The following diagram shows a chain tied together by the DARE aggregator:

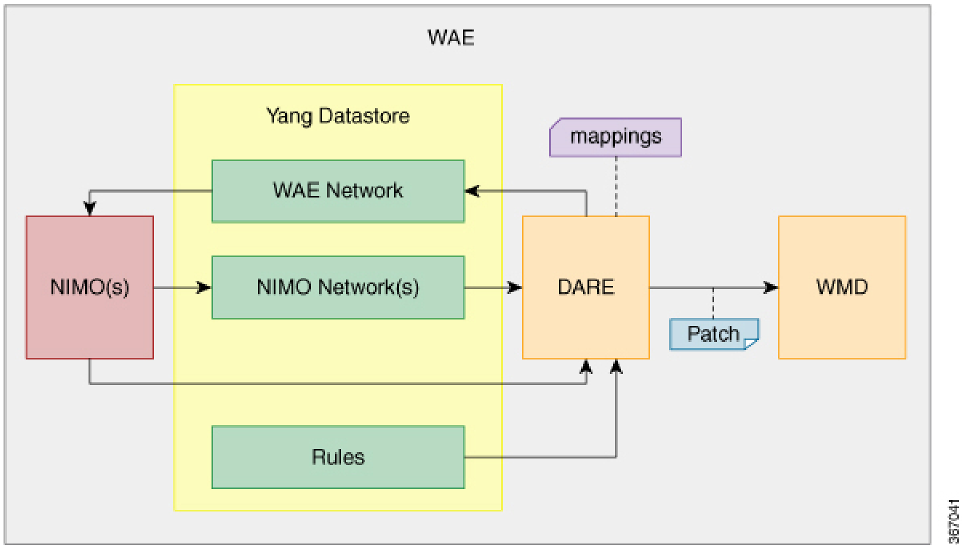


As shown in the preceding diagram, a model building chain can branch, allowing for independent, parallel model building tasks. The result is that each branch contains different model information. In the preceding example, one branch ends up with LSPs and traffic measurements; the other branch ends up with a model that can be better visualized.

The DARE aggregator is a WAE component that brings together various model building chain branches, selecting model information from each of them, and consolidating the information in a destination model. The preceding example is configured to look at all models in the chain. With the configuration shown, the aggregator immediately picks up changes to the topology model. Without a connection, the changes would have to propagate up the model building chain branches before being processed to the top-level model. The aggregator routes changes to its destination model to the correct downstream NIMO. In the preceding example, if you create an LSP at the top-level model, the aggregator forwards that change to the LSP model.

Delta Aggregation Rules Engine

Delta Aggregation Rules Engine (DARE) is a WAE component that brings together various model building chain branches, selects model information from each of them, and consolidates the information in a destination model (final network model). It collects changes to the NIMO models by subscribing to notifications and providing an API for NIMOs to publish changes directly to. These changes are aggregated based on configured rules. In addition, the changes are sent in parallel to the WAE Model Daemon (WMD) in the form of a patch. DARE stores its own state required for the aggregation in various maps on the file system.



Note Since DARE works and is based off of changes, it should be configured before changes are made to NIMO models.

For information on how to configure the aggregator to use DARE, see NIMO Collection Consolidation.

WAE Modeling Daemon (WMD)

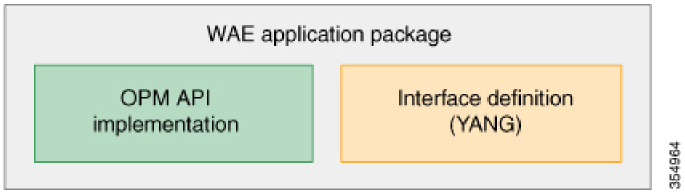
WMD receives changes from DARE, incorporating scheduled NIMO runs as well as reactive updates from the XTC Agent to Patch module. It also schedules insertions of measured traffic updates into the in-memory model from the traffic poller NIMO. All updates are consolidated into a near real-time Master Model of the network. WAE applications (described in the next section) are able to connect to WMD and gain access to a copy of this near real-time model in order to leverage WAE OPM API functionality.

For information on how to configure WMD, see Configure the WAE Modeling Daemon (WMD).

WAE Applications

WAE provides a flexible and powerful application development infrastructure. A simple WAE application consists of:

- The application interface, defined in a YANG model. This interface usually includes RPCs and data models. The YANG models can, if necessary, extend the WAE network model, adding new data types.
- The application logic, implemented using the OPM API.



Because WAE automatically generates APIs from YANG definitions, a WAE application has its APIs automatically exposed. A WAE application is, in a sense, a seamless extension of WAE functionality.

Bandwidth on Demand Application

The Bandwidth on Demand (BWoD) application utilizes the near real-time model of the network offered by WMD to compute and maintain paths for SR policies with bandwidth constraints delegated to WAE from XTC. In order to compute the shortest path available for a SR policy with a bandwidth constraint and ensure that path will be free of congestion, a Path Computation Element (PCE) must be aware of traffic loading on the network. The WAE BWoD application extends the existing topology-aware PCE capabilities of XTC by allowing delegation of bandwidth-aware path computation of SR policies to be sub-delegated to WAE through a new XTC REST API. Users may fine-tune the behavior of the BWoD application, affecting the path it computes, through selection of application options including network utilization threshold (definition of congestion) and path optimization criteria preferences.

For information on how to configure the BWoD application, see Bandwidth on Demand Configuration Workflow.

Bandwidth Optimization Application

The Bandwidth Optimization application is an approach to managing network traffic that focuses on deploying a small number of LSPs to achieve a specific outcome in the network. Examples of this type of tactical traffic engineering are deploying LSPs to shift traffic away from a congested link, establishing a low-latency LSP for priority voice or video traffic, or deploying LSPs to avoid certain nodes or links. WAE provides the Bandwidth Optimization application to react and manage traffic as the state of the network changes.

For information on how to configure the Bandwidth Optimization application, see Bandwidth Optimization Application Workflow.

Cisco WAE Interfaces

Cisco WAE has three interfaces that you can use to configure your network model:

WAE UI

The WAE UI provides an easy-to-use interface that hides the complexity of creating a model building chain for a network. The WAE UI combines the configuration of multiple data collections under one network and can produce a single plan file that contains the consolidated data. However, there are certain operations that cannot be performed with the WAE UI. Any configurations done using the WAE Expert Mode or CLI may not appear in the WAE UI configuration screens. See Network Model Configuration—WAE UI and Important Notes.

Expert Mode

The Expert Mode is a YANG model browser with additional device and service functionality that might not be available in the WAE UI. Users might prefer to use the Expert Mode over the WAE CLI because all options for each operation are visible in the Expert Mode. See Network Model Configuration—Expert Mode.

WAE CLI

The WAE CLI is the interface in which the user responds to a visual prompt by typing a command; a system response is returned. It is the bare-bones interface for all WAE configurations. Operations available in the Expert Mode are also available in the CLI. See Network Model Configuration—WAE CLI.

Network Model Creation Workflow

The following is a high-level workflow on how to configure individual network models. The detailed steps differ depending on what type of interface you use (Expert Mode, WAE UI, or WAE CLI).

If you plan to run multiple NIMOs and consolidate the information into one final network, do not run collections until after you have set up the aggregator NIMO. For more information, see NIMO Collection Consolidation.

1. Configure device authgroups, SNMP groups, and network profile access.
2. (Optional) Configure agents. This step is required only for collecting XTC, LAG and port interface, or multilayer information.
3. Configure a network with a basic topology collection.
4. Run the collection.
5. Configure additional network collections.
6. (Optional) Schedule when to run collections.
7. Configure the archive file system location and interval at which plan files are periodically stored.
8. (Optional) View plan files in WAE applications.