P2P Network Basics 🌐

The `P2PNet` library provides the core functionality for building and managing a peer-to-peer network. It includes classes and methods for peer discovery, connection management, data exchange, and network routines. This document provides a broad overview of the peer network basics.

Initialization

The `PeerNetwork` class is the main entry point for initializing and managing the peer-to-peer network. It sets up the necessary configurations and services required for network operations.

1. **Configuration**: The application reads configuration settings from the provided parameters. This includes settings for designated ports, inbound peer acceptance, and trust policies.
2. **Logging**: Logging is configured to use a plain text format and is activated to capture important events and errors.
3. **Local Address Loading**: The application scans all network interface devices and collects essential information needed for the peer network, such as public IP addresses.

Operation

The `PeerNetwork` class operates by providing several key functionalities:

1. **Peer Discovery and Connection**: The peer network supports both LAN and WAN peer discovery. It uses broadcasting and multicasting to discover peers within the network and establishes connections with them.
    - **LAN Discovery**: The application can broadcast and discover peers within the local network.
    - **WAN Discovery**: The application can discover peers over the wide area network using designated ports and WAN components.
2. **Peer Management**: The peer network manages a list of known peers and active peer channels. It supports adding, removing, and elevating peer permissions.
    - **Known Peers**: A list of peers that have been discovered and are known to the network.
    - **Active Peer Channels**: A list of active peer channels that are currently connected and exchanging data.
3. **Data Exchange**: The peer network supports data exchange between peers using peer channels. It handles sending and receiving data, as well as processing incoming packets.
    - **Peer Channels**: Represents a communication channel between two peers. It handles the sending and receiving of data packets.

Routines

The `NetworkRoutines` class provides a mechanism for managing network routines. Routines are tasks that run at specified intervals to perform various network-related operations.

1. **Routine Management**: The `NetworkRoutines` class manages a dictionary of routines and provides methods for adding, starting, stopping, and setting the interval of routines.
    - **Default Routines**: The application initializes with default routines, such as rotating broadcast ports and discerning peer channels.
    - **Custom Routines**: Users can add custom routines to perform specific tasks.

Routines are accessed using their `RoutineName` property. This is automatically handled when they are added as network routines.

Trust Policies

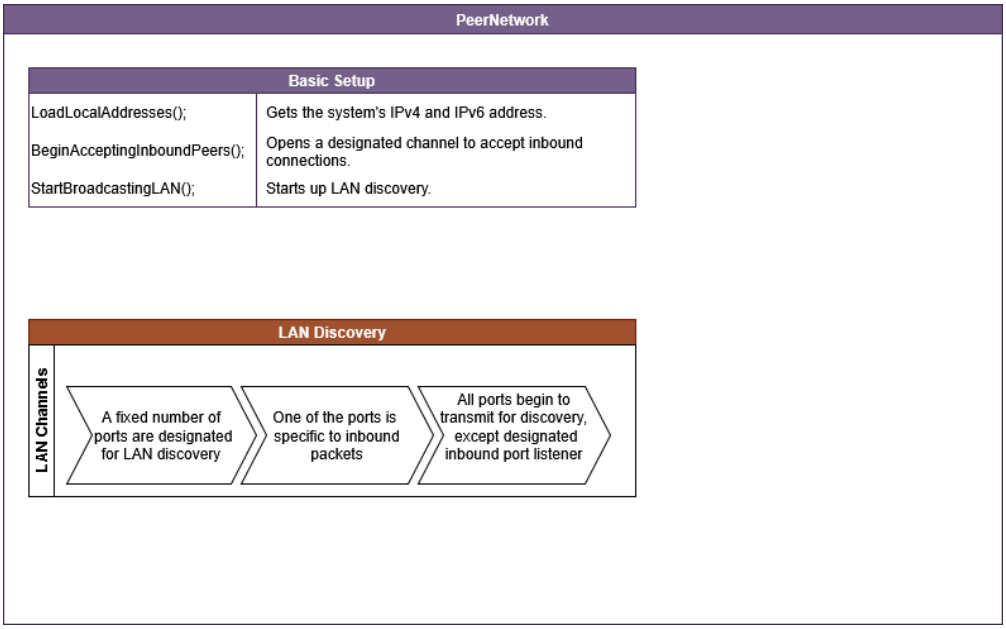
The `PeerNetwork` class includes trust policies for managing incoming peer connections and bootstrap connections.

1. **Incoming Peer Trust Policy**: Handles trust and permissions for incoming peer connections. It supports different modes for handling inbound peers, such as queue-based and event-based.
2. **Bootstrap Trust Policy**: Handles trust and permissions for bootstrap connections. It supports different trust policies, such as trustless and authority-based connections.

Diagrams

To supplement the information visually, the following diagrams are provided:

1. **Peer Network Architecture**: Shows the overall architecture of the peer network, including its interaction with peers and network routines.
2. **Peer Discovery Flow**: Illustrates the flow of peer discovery, from broadcasting to establishing connections and managing peer channels.



subject to change

Namespace P2PBootstrap

Classes

GlobalConfig

Program