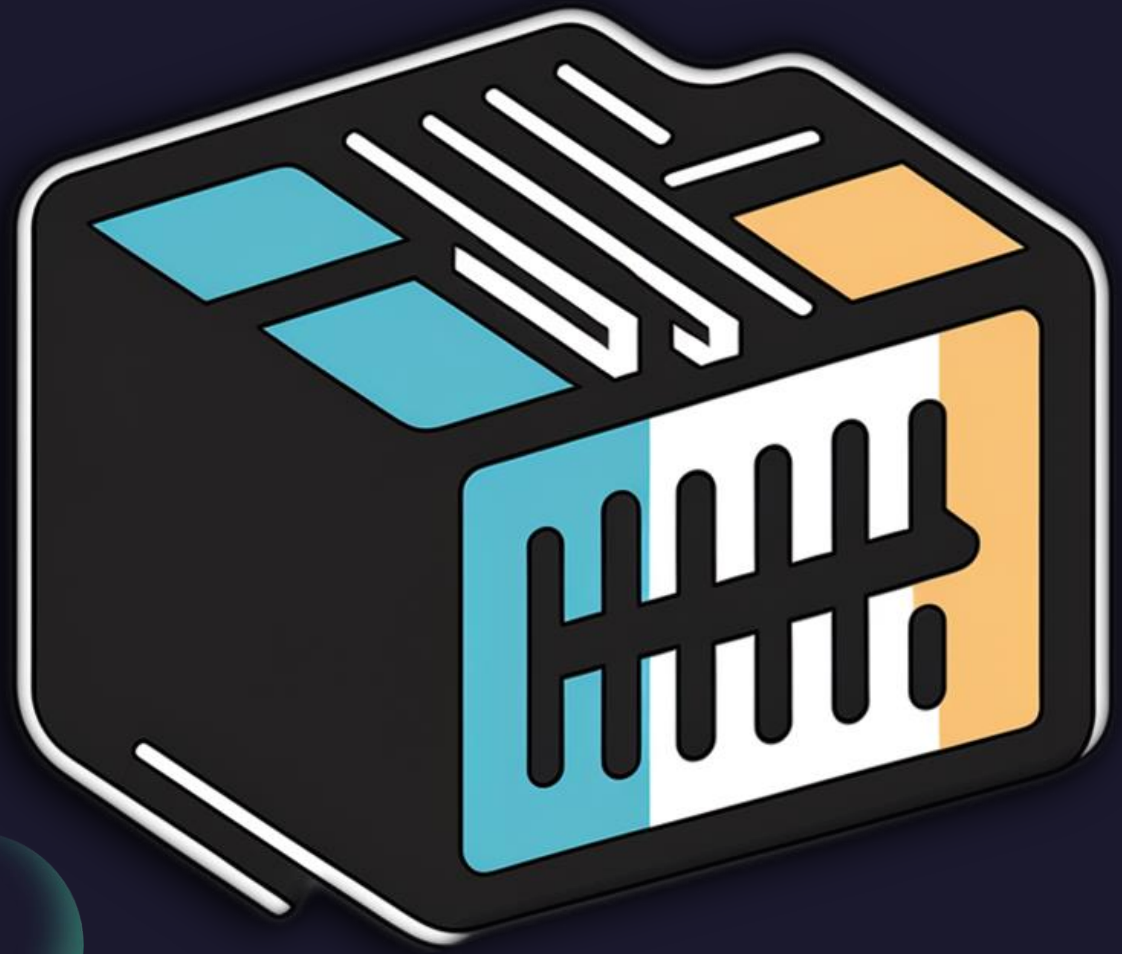
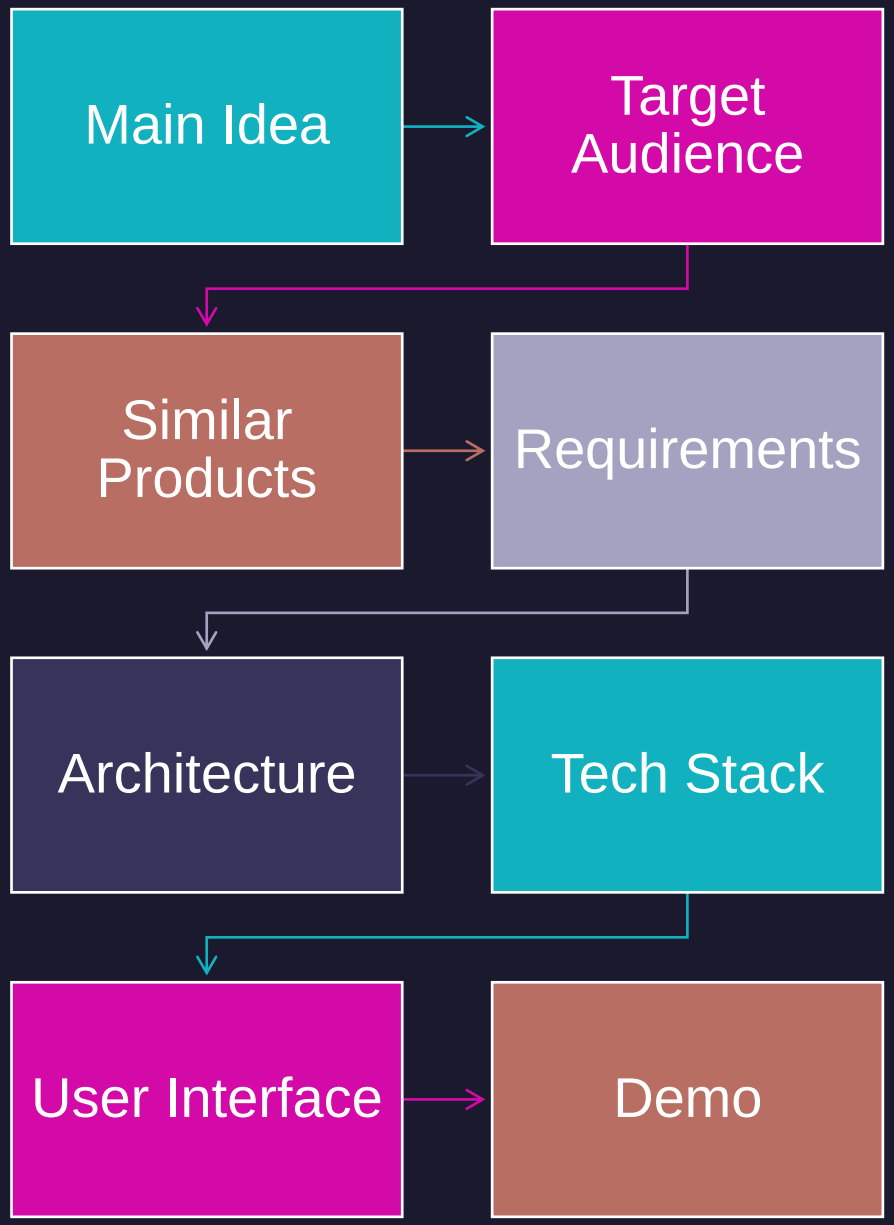
E-Jam

Distributed System for Testing, Monitoring, and Debugging a Network Switch



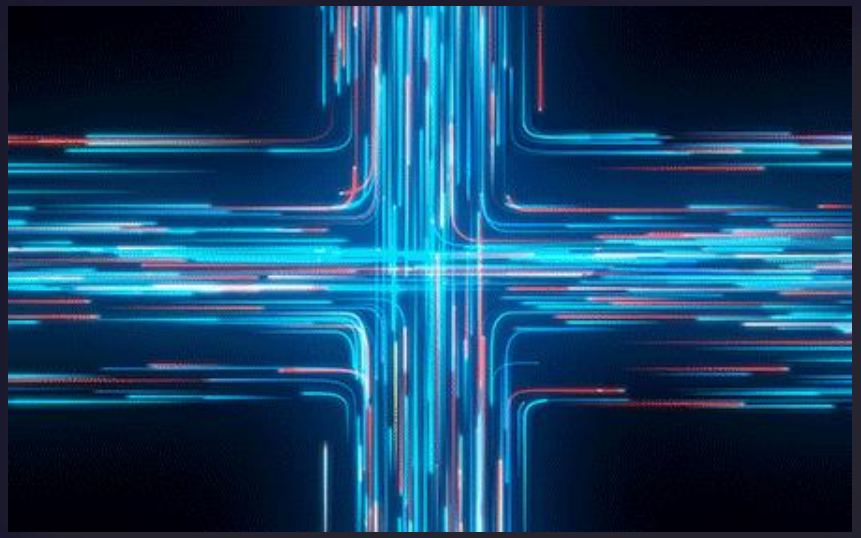
Sponsored By SIEMENS

Agenda


Main Idea
Target Audience
Similar Products
Requirements
Architecture
Tech Stack
User Interface
Demo

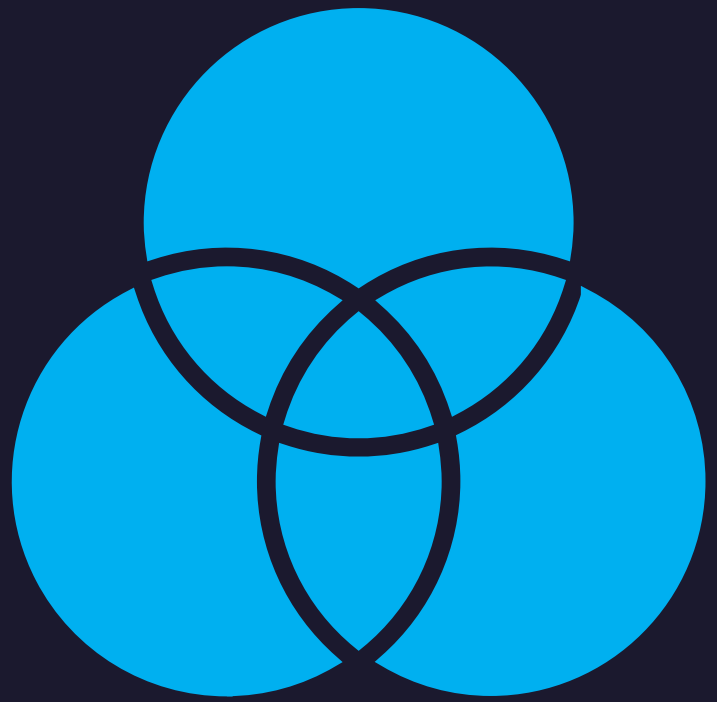
Main Idea

- The network industry's rapid **pace of development** has led to an exponential growth in data generation and processing within switches.
- It is crucial to ensure that switches maintain specific **thresholds** in terms of throughput, latency, and packet loss.
- Network administrators require robust **monitoring** tools to assess **switch performance**, including **metrics** like total packets forwarded and traffic status.



Our Solution

- E-Jam is a comprehensive platform designed to support the **lifecycle** of networking devices and switches.
- Essential facilities provided by E-Jam ensure **seamless** implementation, optimal performance, and adherence to industry **standards**.
- E-Jam offers tools for **verifying**, **testing**, and **debugging** network switches through software, supporting the design lifecycle of switches.



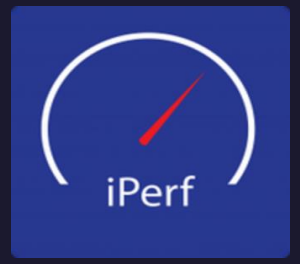
Target Audience

- E-Jam proves great benefit if used in the process of **debugging** and **stress-testing** the network inside a commercial environment.
- Engineers of an **IT division**, **network administrators**, switch **vendors** and **manufacturers** at any business are the main stakeholders of this project.

Similar Products

IPerf

An open-source software developed by ESnet / Berkeley National Lab to measure bandwidth of TCP and UDP on IP networks



	E-Jam	**IPerf**
Pros	- Measures additional statistics (dropped packets, throughput).	- Measures bandwidth and packet loss
	- Provides more visual reporting of data and ease of access	- Allows tuning of TCP parameters and UDP bandwidth
	- Flexible testing using no specific network layer protocol	- Cross-platform compatibility
	- Allows custom tests for simulating real-world environments	
Cons	- Works only on Linux devices For the System API	- Provides limited reporting capabilities
		- Limited to specific network layer protocols

SolarWinds Network Bandwidth Analyzer Pack

A commercial software by SolarWinds for active performance and troubleshooting and networks

	E-Jam	**SolarWinds**
Pros	- Provides the facility for custom tests	- Identifies bandwidth hogs
	- Offers comprehensive simulation of real environments	- Provides monitoring and analysis capabilities
		- Analyzes traffic routes and provides hop-by-hop analysis
	- Provides a sleek modern extendable graphical interface	
Cons	- Limited compatibility (Linux devices only)	- Limited to monitoring and analysis capabilities only

Functional Requirements

Verifying the traffic generated passing through the switch.

Custom tests for measuring switch performance under user-defined conditions.

Pre-set tests to automatically configure streams.

Detailed Logging for events.

Live graphs for the ongoing test.

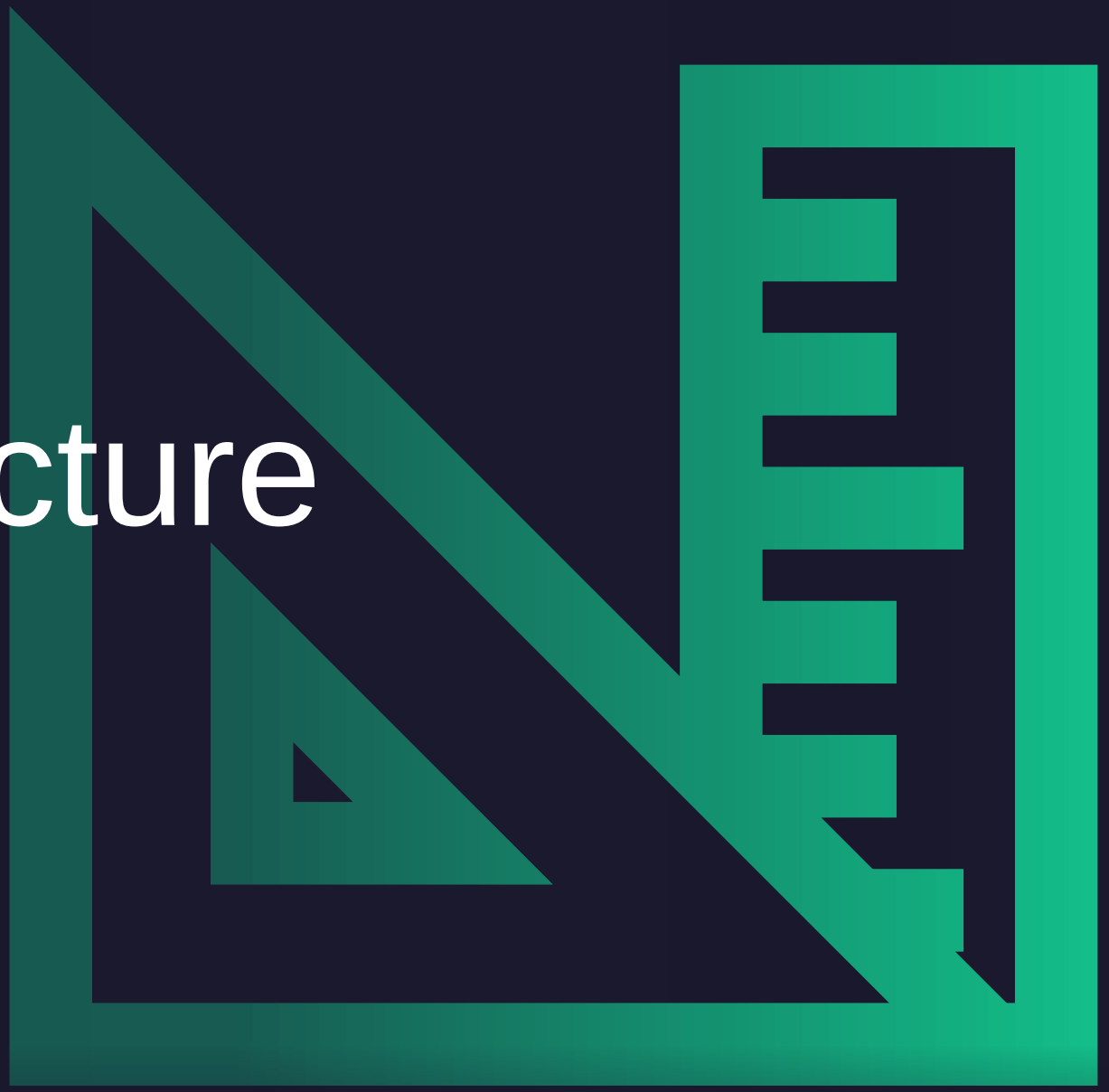
Maintain results of previous tests and **export** the provided results.

Non-Functional Requirements

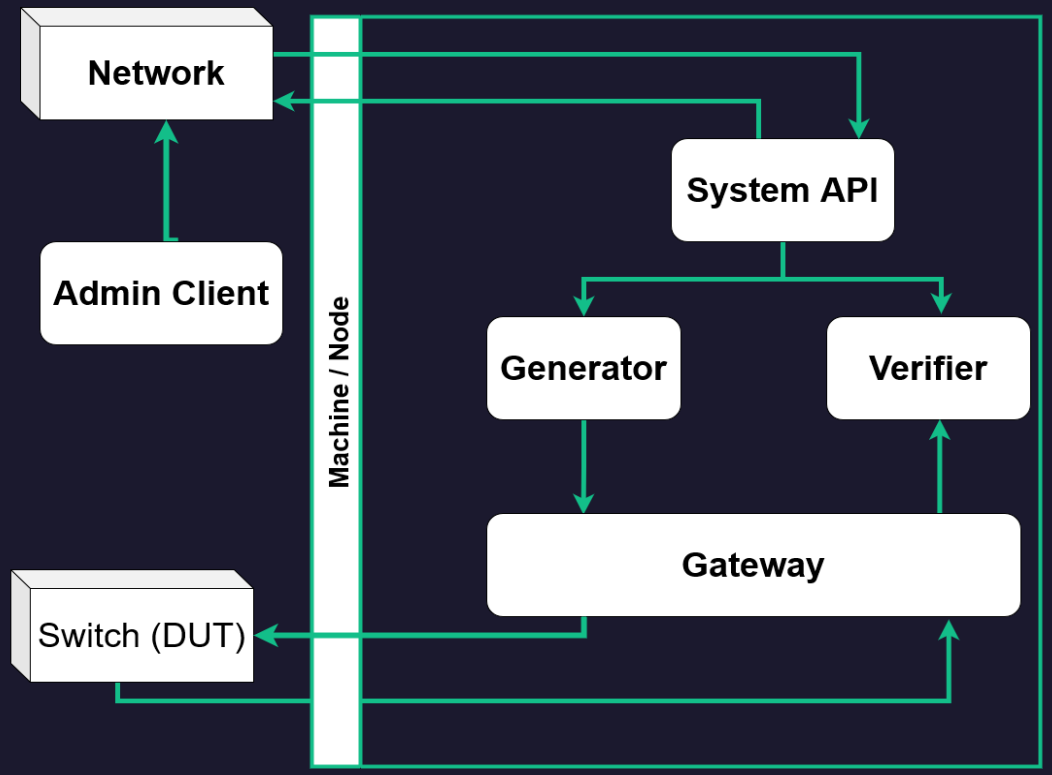
While the core system operates in a **Linux environment**, the User Interface (UI) component of E-Jam is designed to be **cross-platform**.

E-Jam is designed to be **user-friendly** and intuitive, even for individuals with **basic knowledge** of switches and their functions.

Architecture



[Centralising controls]



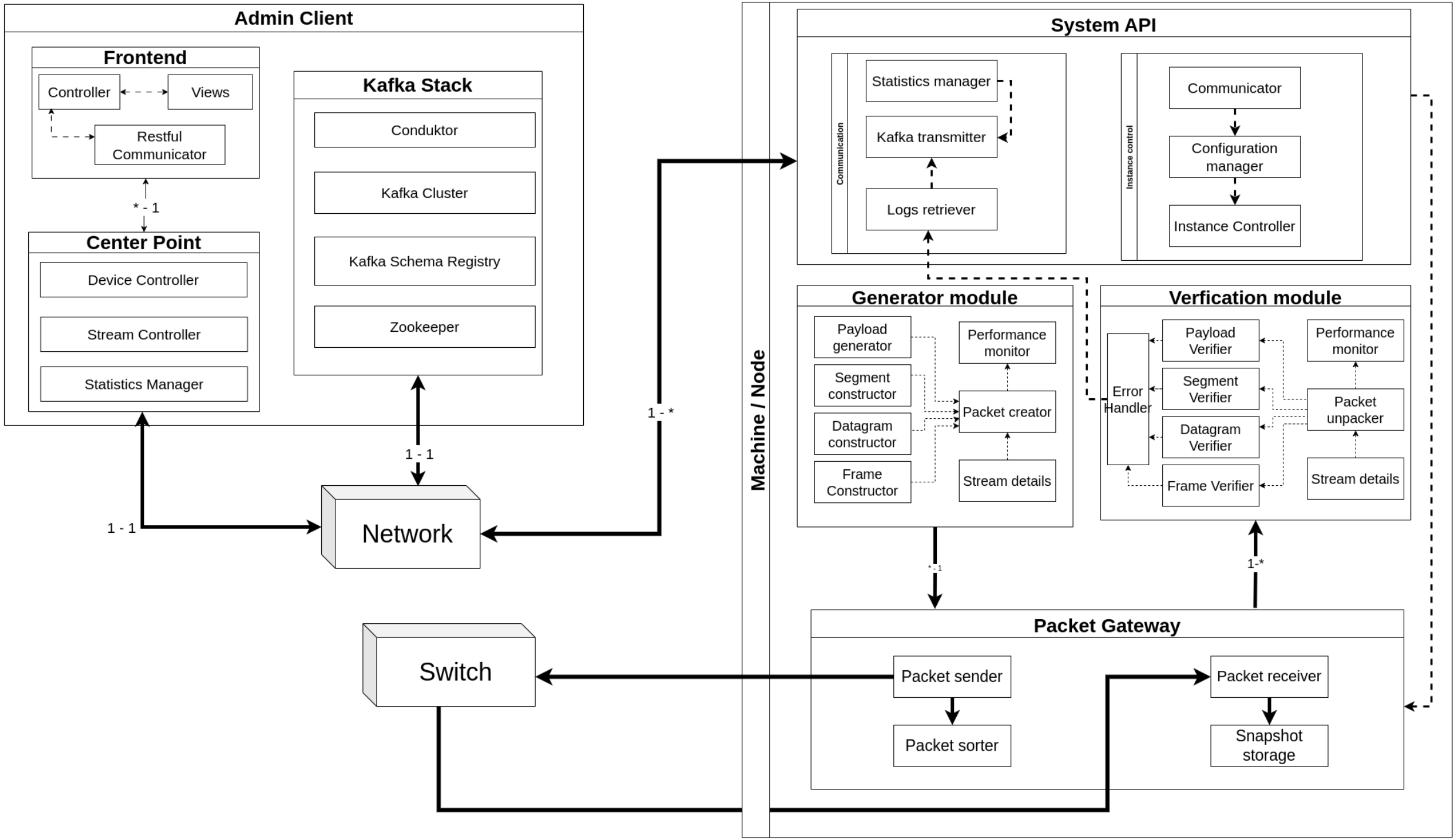
- A test is defined by a set of data streams with a certain packet configuration.
- In a stream, groups of devices are specified as Generators and other groups are specified as Verifiers.
- The software can operate on one or more devices (nodes), ideally on multiple devices (to provide more power for tests). All nodes must be connected through Ethernet ports to the switch under test (Device Under Test DUT).
- The system provides an admin which has a GUI where a user can start, control, and monitor test procedures and idle states.

The Admin Client Submodules

Center Point

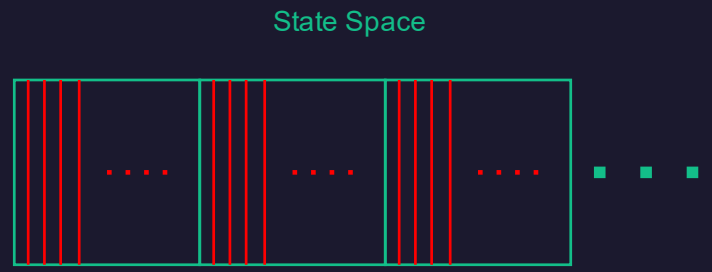
Admin Client API

Admin Client GUI


Admin Client
Frontend
Controller
Views
Restful Communicator
* - 1
Center Point
Device Controller
Stream Controller
Statistics Manager
Kafka Stack
Conduktor
Kafka Cluster
Kafka Schema Registry
Zookeeper
1 - 1
1 - 1
1 - *
Network
Switch
Machine / Node
System API
Communication
Statistics manager
Kafka transmitter
Logs retriever
Instance control
Communicator
Configuration manager
Instance Controller
Generator module
Payload generator
Segment constructor
Datagram constructor
Frame Constructor
Performance monitor
Packet creator
Stream details
Verfication module
Error Handler
Payload Verifier
Segment Verifier
Datagram Verifier
Frame Verifier
Performance monitor
Packet unpacker
Stream details
* - 1
1-*
Packet Gateway
Packet sender
Packet sorter
Packet receiver
Snapshot storage

Generation of Parallel Random Streams

- We need multiple random streams for parallel generators to be truly random.
- We Implement this by using pseudo random number generators which have a facility to jump ahead in state (F2-Linear Random Number Generator).
- This allows us to partition the state space amongst different generators, and to further partition the larger partitions into smaller partitions for each packet.



Sequencing

- We detect out of order packets by attaching a sequence number to each packet in a stream (per generator).
- We maintain a small number of previous missing packets to check if they were out of order.
- After arrival of a certain number of packets with a greater sequence number, the packet is considered dropped.

Tech Stack

Flutter

Docker

Conduktor

Kafka

Rust

Spring boot

Java

C++

Linux

Graphical User Interface

Dashboard


E-Jam
Dashboard
Horizo...
Right-...
Manda...
Mand...
Versat...
Syner...
Custo...
Netw...
Proac...
Orga...
Versa...
Unive...
Assi...
Pre-e...
Expan...
Digitiz...
User-c...
Opera...
Re-en...
Mana...
Netw...
Balan...
De-en...
Syner...
Visio...
Right...
Fund...
Multi...
Orga...
Balance...
Secur...
Right-...
Persev...
Clone...
Face t...
Enter...
Organ...
Re-en...
Reacti...
Rever...
Intuiti...
Orga...
Oper...
Diver...
Futur...
Open-ar...
Exclusive ...
Sy...
Enhance...
Team-ori...
Performance
AC Pkt
Upload
Download
Error Rate
Elements
50
10
11
2497
2468
100
16
18
13
13
Progress
59.23%

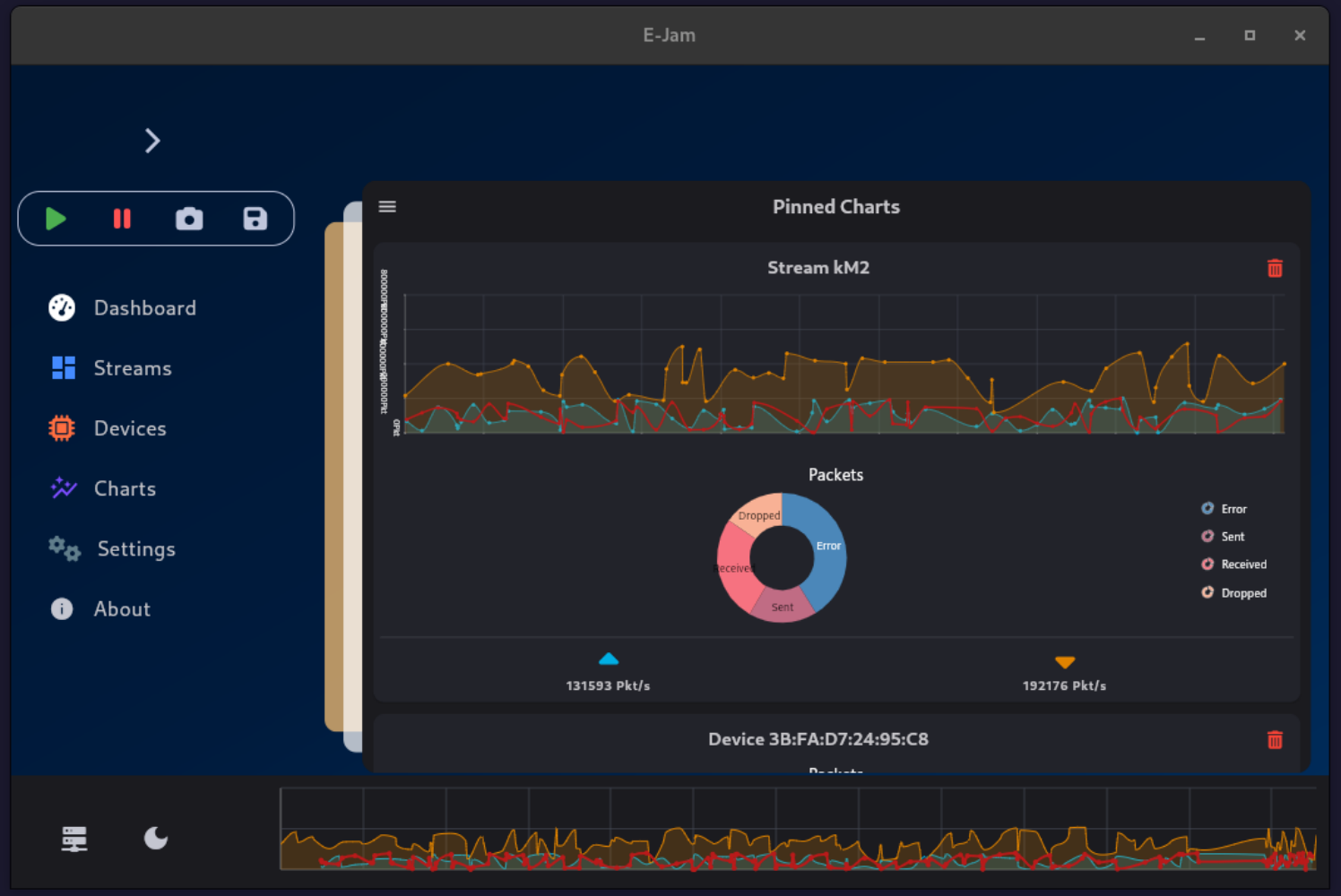
Stream View


E-Jam
Stream wDc
Triple-buffered
wDc
Generation Seed
8949877238746561743
Delay
1344...
Duration
8870663...
Inter Fr...
13518390...
Packets
14604265856...
Frame Size
128572531475580...
Payload Len...
240
Payload Type
Ipv6
Burst Length
91802183816...
Burst Delay
168480233051672...
Flow Type
BackToBack
TLP Type
UDP
Packets
Dropped
Error
Sent
Received
Error
Sent
Received
Dropped
Devices
Queued
Stopped
Completed
Running
Failed
Failed
Running
Completed
Queued
Stopped
0
10
20
30
40
50
60
70
80
90
100

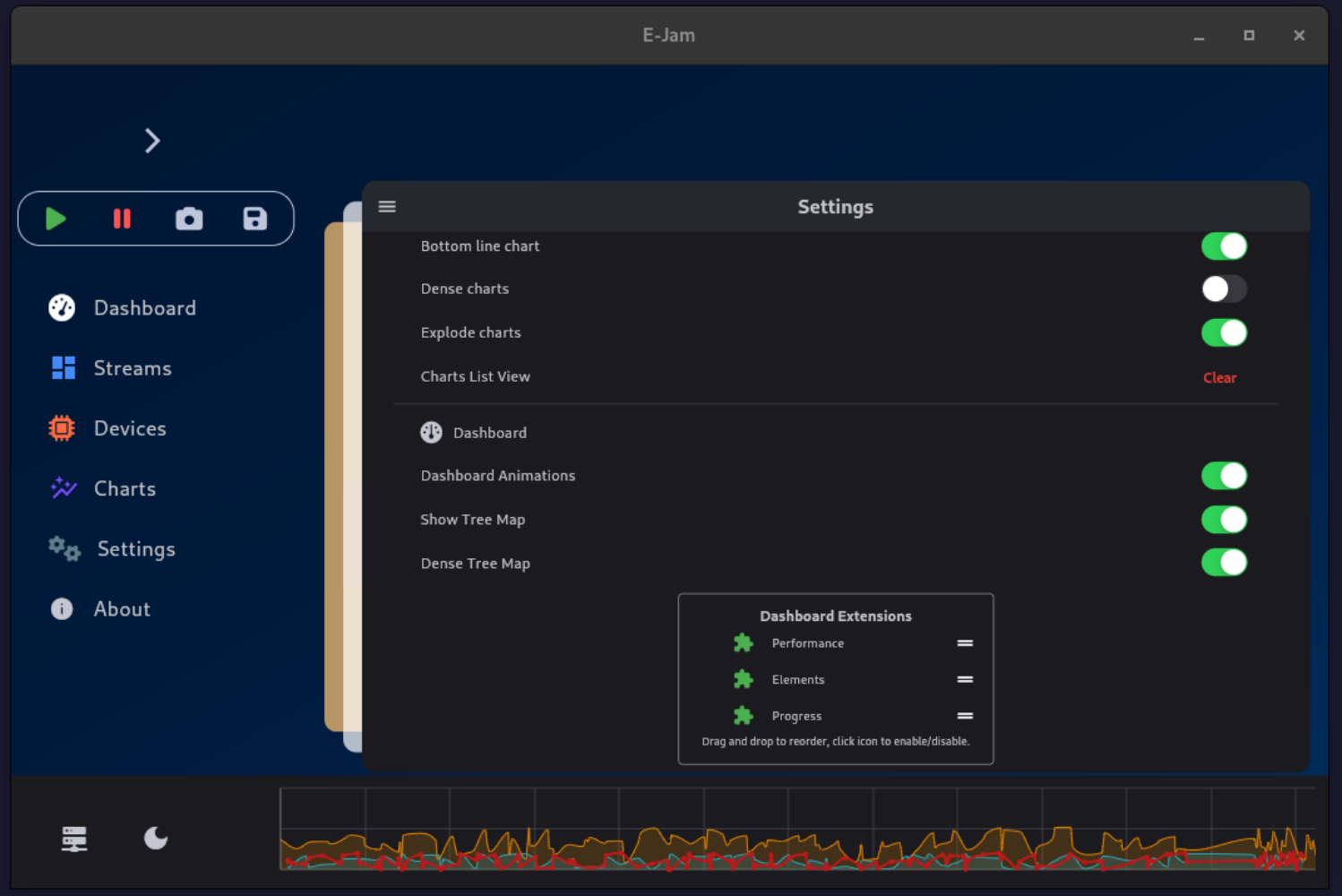
Device View

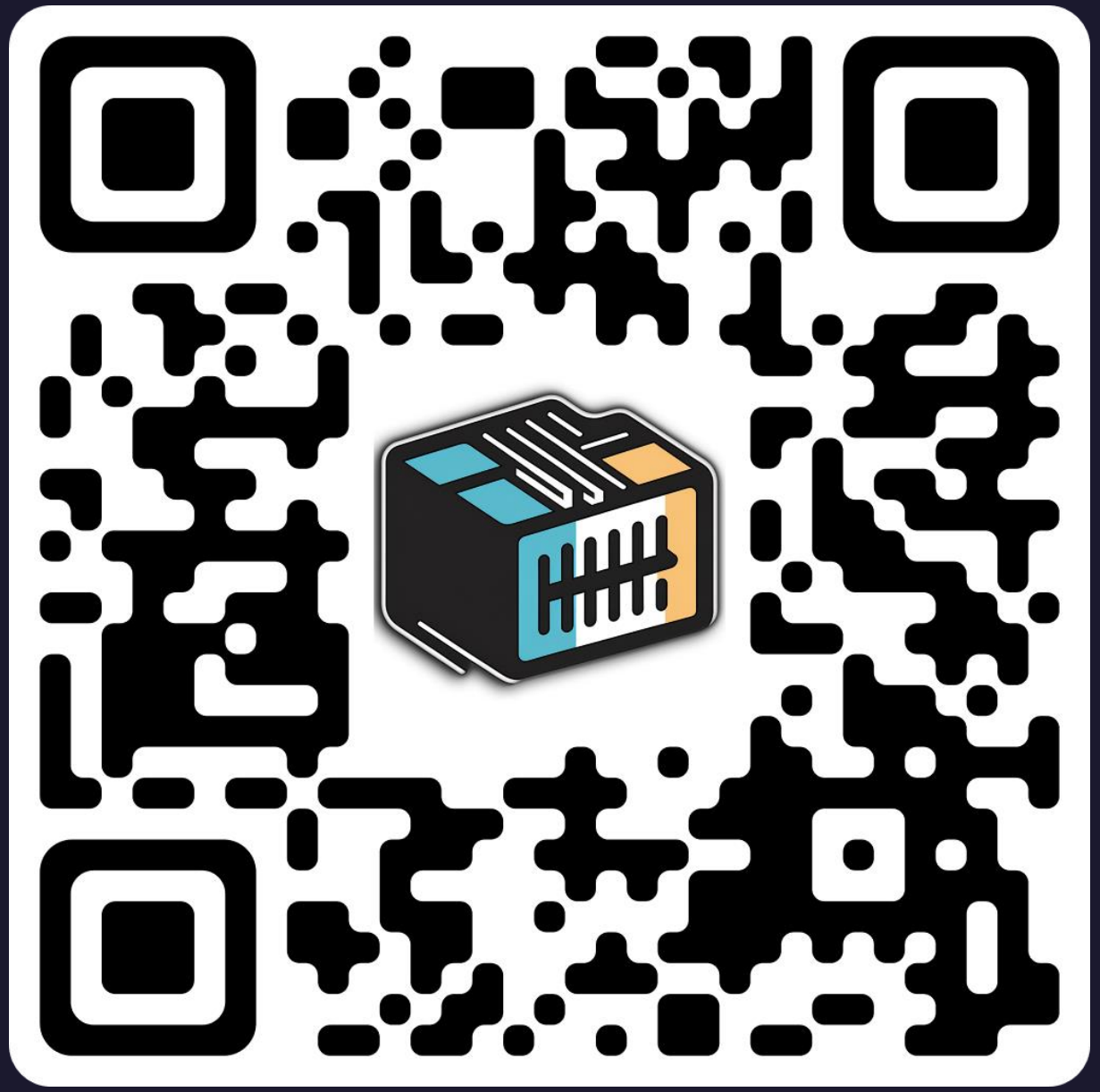

E-Jam
Dashboard
Streams
Devices
Charts
Settings
About
Organized linux cre...
Packets
Dropped
Error
Received
Sent
Error
Sent
Received
Dropped
46
48
Organized linux creola_ipsum
169.181.29.232 : 9754
Konopelski ton
187794 Pkt/s
181605 Pkt/s
Fundamental lin...
6.56.160.2
Borer bury
Online
Idle
Right-sized lily_et
213.152.233.14
Veum furt

Pinned Charts



Setting View





Video Demo

This video will be highlighting some key features and functions of the project and explaining how they work.

Thank You