

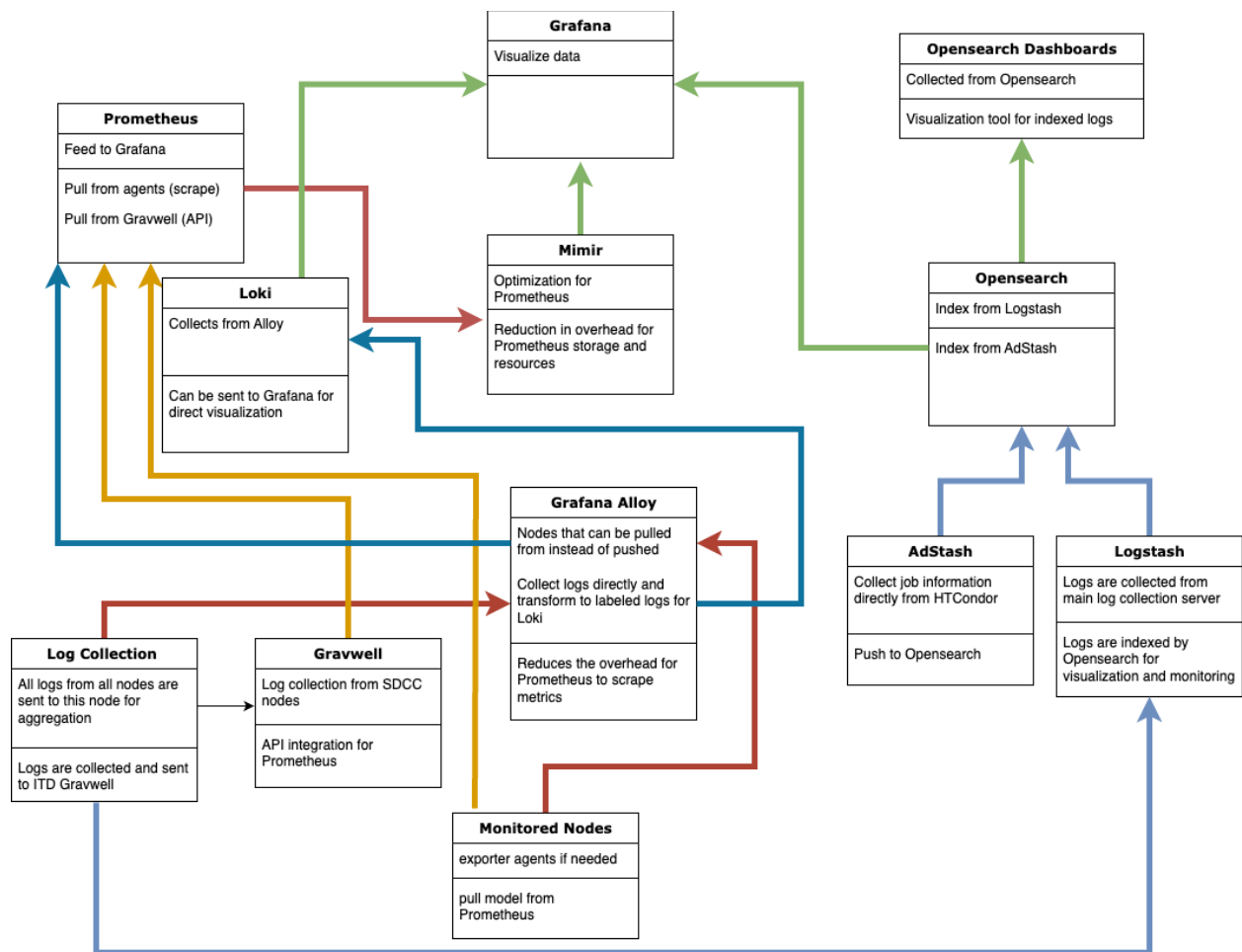
Current production:

- SDCC is currently maintaining and operating a monitoring stack of:
 - Elasticsearch - Data aggregation/indexing/querying
 - Logstash - Log collection/ingestion
 - Kibana - Log analysis/anomaly detection
 - Grafana - Visualization
- The current production setup is monitoring ~1k+ hosts
- An Opensearch cluster with 4 nodes has been added in parallel to this setup for testing purposes

In development:

- SDCC is currently working on optimizing our monitoring setup:
 - Prometheus is being added as an additional service for anomaly detection, possible integration with Gravwell, and to expand integration with various monitoring agents
 - Opensearch is replacing Elasticsearch, Elasticsearch is no longer open-source as of version 7.11.
 - Replacing Elasticsearch with Opensearch will allow SDCC to continue receiving updates as Opensearch is a fork of 7.10 Elasticsearch.
 - Opensearch is compatible with many of the current Elasticsearch API's.
 - Machine learning and anomaly detection (machine learning is a paid feature in Elasticsearch.)
 - Opensearch-Dashboards to replace Kibana for better integration with Opensearch
 - Many Kibana plugins are now unsupported with Opensearch
 - Opensearch-Dashboards includes security, role-based access controls, anomaly detection
 - Future-proofing for Opensearch updates
 - Gravwell integration using Prometheus
 - Prometheus is able to utilize the Gravwell API which is currently collecting logs from many sources
 - Further aggregation with Gravwell can provide more data for better predictive modeling
 - Grafana Alloy:
 - Alloy uses push-based metrics collections instead of pull-based which will reduce the server load for Prometheus
 - Improves scalability for Prometheus
 - Supports Prometheus metrics and Loki logs
 - Loki:

- Uses the same labels as Prometheus, compatibility between metrics and logs
 - Optimized for storing and querying recent logs efficiently
 - Label only allows for quickly searching logs to troubleshoot
- Grafana Mimir to support Prometheus:
 - Long term storage allows Mimir to keep metrics for analyzing historical trends
 - Provides sharding and replicating across multiple nodes, assists with scaling as Prometheus is run as a single node
 - Multiple replicas ensure no single point of failure
 - Optimized storage with compression
 - Integration with Grafana
- AdStash:
 - Performance optimized monitoring of HTCondor nodes
 - Monitor job usage and alerts
 - Integration with Opensearch
- Logstash will continue to be used for metrics collection, this may be replaced in the future with Gravwell and Prometheus integration
- Grafana will continue to be used for visualization and Alloy/Mimir usage
- Monitoring will be scaled to meet production needs:
 - Hardware and storage will be needed to support this as it will eventually become a production system



- Nodes can be pulled directly from Prometheus but to reduce server overhead we can use Alloy to collect and then push to Prometheus.
- Alloy can also collect logs and label for Loki to provide a quicker reference for troubleshooting
- Gravwell integration can create a pathway to bypass the need for a separate Logstash configuration reducing storage costs for logs, OpenSearch would still be needed however for AdStash/HTCondor with a smaller storage pool
- Mimir provides a storage optimization for Prometheus and further reduces the needed resources for keeping this data
 - Ability to scale over multiple nodes
 - Ability to allow multiple Prometheus instances to write to it
 - Replicates data to ensure no single point of failure
 - Faster queries than Prometheus alone