OSS Tools
for System Management

Sanjula Ganepola, IBM
Software Developer
sanjula.ganepola@ibm.com

Agenda

IBM i
continuous innovation
continuous integration

- Current State of System Management on IBM i
- Operational Monitoring with Prometheus
- Data Visualization with Grafana
- Event Monitoring with Manzan
- Manzan + AI

IBM i
continuous innovation
continuous integration

Current State of System Management on IBM i

What tools are you using for monitoring your IBM i systems?

IBM i
continuous innovation
continuous integration

1. Dynatrace
2. Nagios
3. Instana
4. DataDog
5. Control4i
6. Syslog Reporting Manager (SRM)
7. Created your own
8. Other

Specialty is a collective disadvantage

IBM i
continuous innovation
continuous integration

1. Dynatrace
2. Nagios
3. Instana
4. DataDog
5. Control4i
6. Syslog Reporting Manager (SRM)
7. Created your own
8. Other

Each solutions has their own…

- Configuration
- Host installation requirements
- Monitoring capabilities
  - Collect system metrics (active jobs, ASP consumption)
  - View sub system information
  - Identify long-running SQL
  - View job queue

Do you use Grafana?

IBM i
continuous innovation
continuous integration

1. Yes, with IBM i
2. Yes, but not with IBM i
3. No, but we want to
4. No, we don’t want to
5. No, don’t know what it is

IBM i
continuous innovation
continuous integration

Operational Monitoring with Prometheus

Prometheus Overview

IBM i
continuous innovation
continuous integration

What is it?

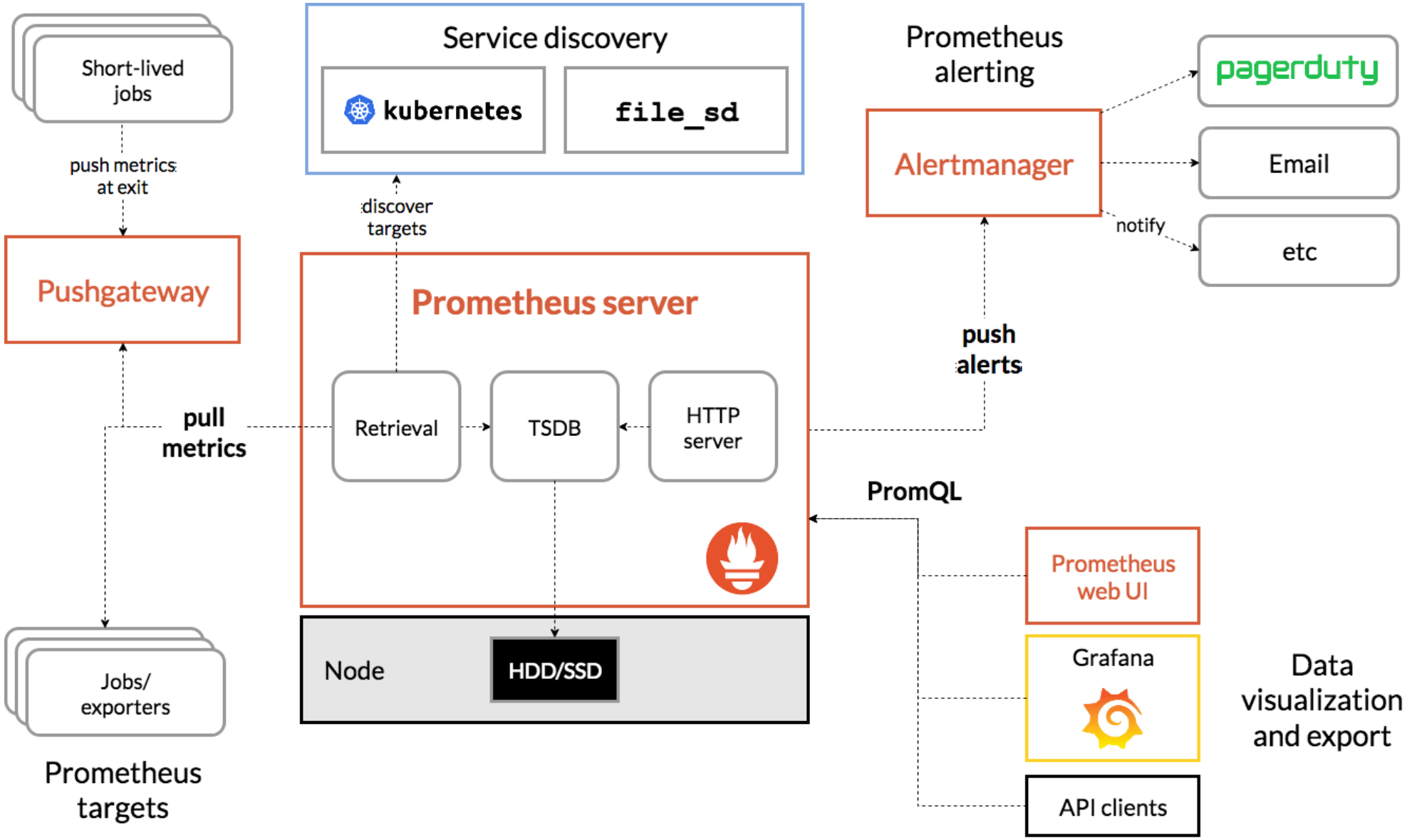
- Leading open-source systems monitoring and alerting toolkit
- Collects and stores metrics as timeseries data

Features

- Multi-dimensional data model with time series data identified by metric name and labels
- Provides a functional query language called PromQL
- No reliance on distributed storage
- Has an alert manager built-in
- Easily paired with Grafana and other monitoring solutions

Prometheus

Prometheus Architecture



Prometheus Data Model

- All data is stored as a time series: `(timestamp, value)`

Ordered by timestamp

Integer or fractional value

- Each time series has a name (metric name)
  - General feature of a system that is measured

- Each time series can have key/value pairs (metric labels)
  - Identifies a particular dimension of the metric

- Notation: `<metric name>{<label name>=<label value>, ...}`
  - `api_http_requests_total{method="POST", endpoint="/messages"}`

- Time series is uniquely identified by metric name + metric label
  - `temperature{city="Toronto"}`
  - `temperature{city="Rochester"}`

PromQL (Prometheus Query Language)

IBM i
continuous innovation
continuous integration

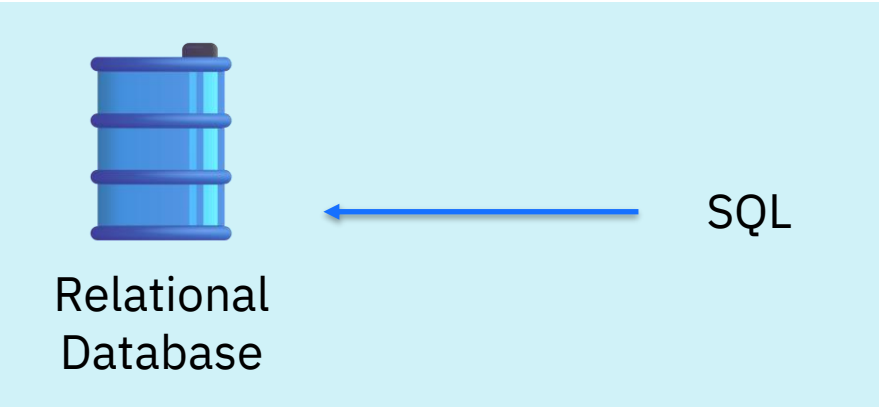
Functional query language that lets the user select and aggregate time series data in real time

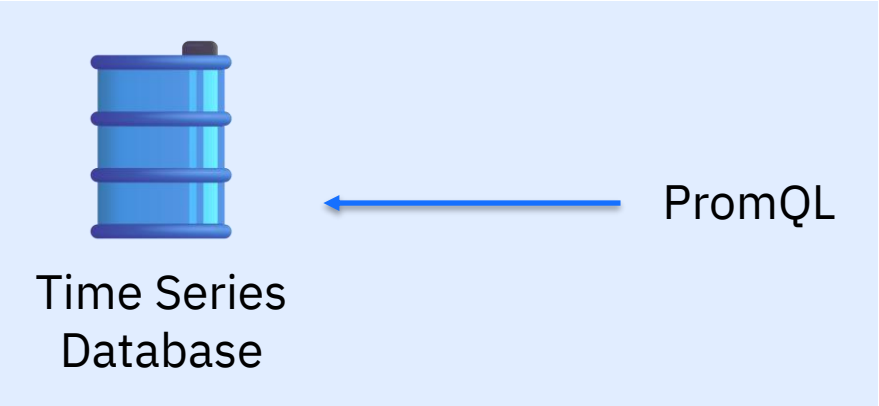
Types:

- Instant query: Evaluated at one point in time
- Range query: Evaluated at equally-spaced steps between a start and an end time

Usage:

- Get/filter metrics we are interested in
- Aggregate metrics
- Build dashboards
- Setup alerts



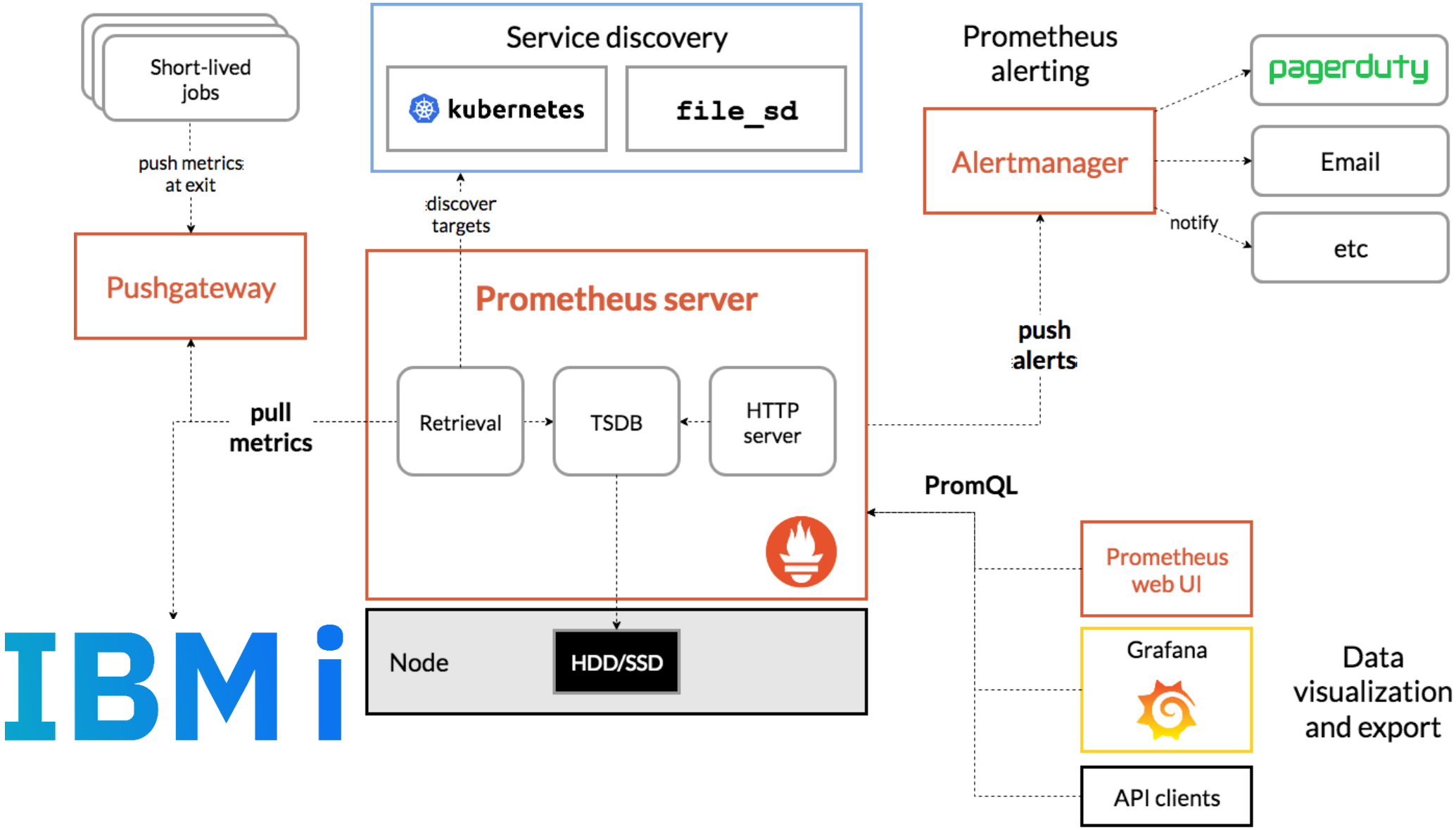


PromQL Basics

IBM i
continuous innovation
continuous integration

SQL	PromQL
select * from http_server_request_count	http_server_quest_count
select * from http_server_request_count where uri="/api/people"	http_server_quest_count{uri="/api/people"}
select * from http_server_request_count where uri="/api/people" and method="GET"	http_server_quest_count{uri="/api/people",method="GET"}
select * from http_server_request_count where status like '2%' or status like '3%' or status like '4%'	http_server_quest_count{status=~"2..|3..|4.."}

How to use Prometheus with IBM i

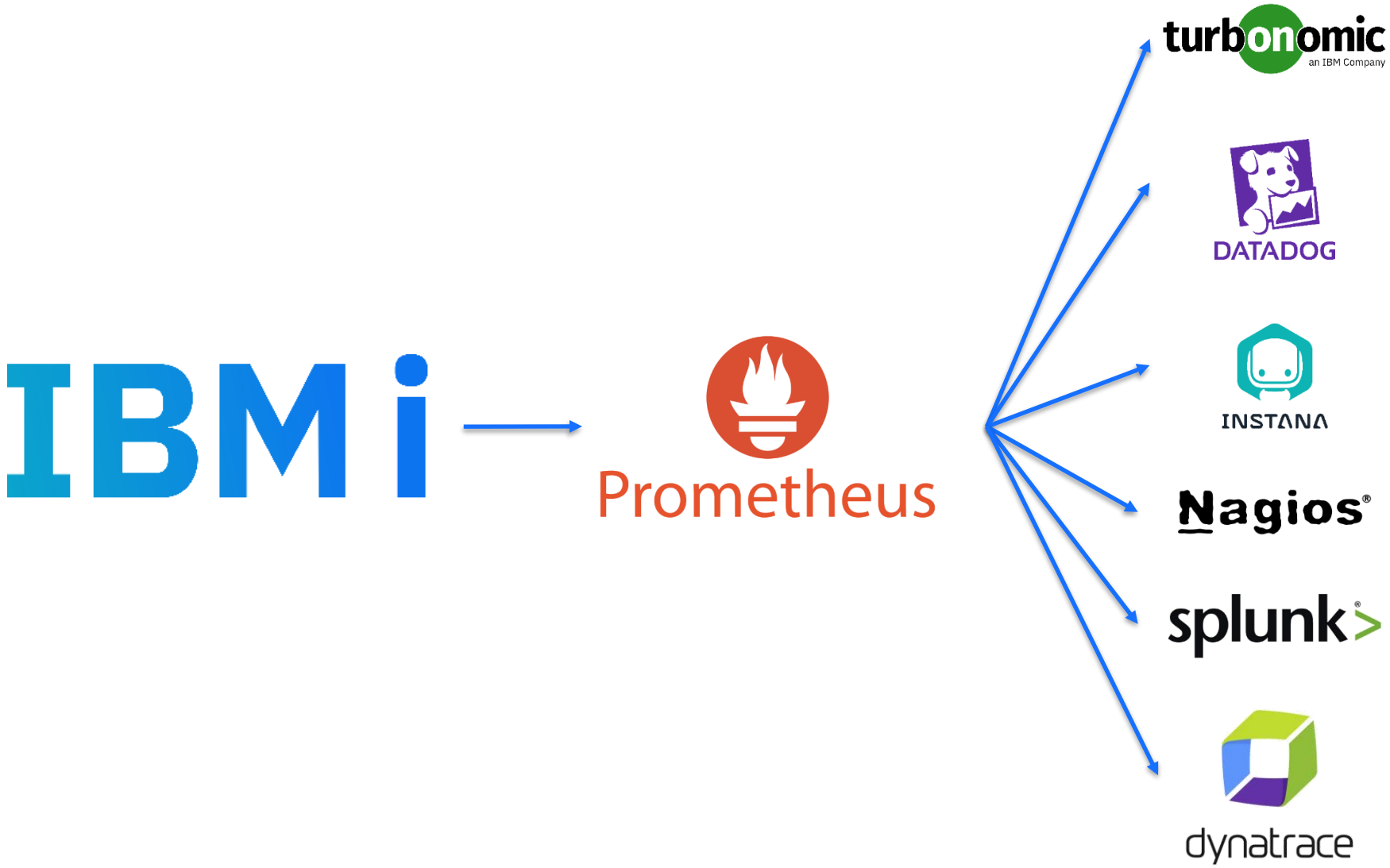


Monitoring IBM i with Prometheus

- Blog post by Jesse Gorzinski: "Monitoring IBM i with Prometheus“ https://techchannel.com/Trends/12/2022/ibm-i-prometheus

- Simplified view
  - Passive exporter running on IBM I
  - Prometheus running on some central location, preferably Docker or Podman
  - Grafana running somewhere, preferably Docker or Podman

Prometheus can be the bridge to other solutions



JDBC Prometheus Exporter

IBM i
continuous innovation
continuous integration

- https://github.com/ThePrez/prometheus-exporter-jdbc
- An interface for passive metric collection which allows Prometheus to scrape it
- Deploys on IBM I
- Exports over 400 metrics
- Customizable metrics with SQL
- Demo: http://ibm.biz/ibmi-prometheus

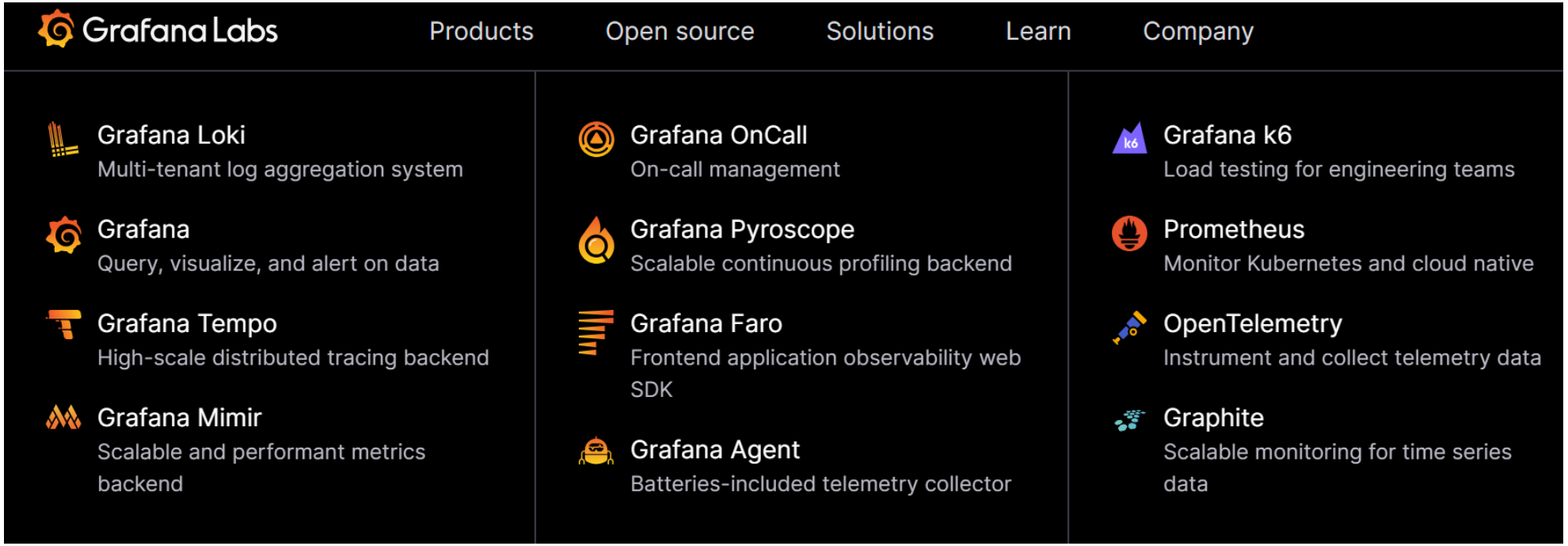
```
{
  "port": 9853,

  "queries": [{
      "name": "System Statistics",
      "interval": 60,
      "enabled": true,
      "prefix": "STATS",
      "sql": "SELECT * FROM TABLE(QSYS2.SYSTEM_STATUS(RESET_STATISTICS=>'YES',DETAILED_INFO=>'ALL')) X"
    },
    {
      "name": "System Activity",
      "interval": 20,
      "prefix": "SYSACT",
      "include_hostname": true,
      "enabled": false,
      "sql": "SELECT * FROM TABLE(QSYS2.SYSTEM_ACTIVITY_INFO())"
    },
    {
      "name": "number of remote connections",
      "interval": 60,
      "sql": "select COUNT(REMOTE_ADDRESS) as REMOTE_CONNECTIONS from qsys2.netstat_info where TCP_STATE
    },
    {
      "name": "Memory Pool Info",
      "interval": 100,
      "multi_row": true,
      "prefix": "MEMPOOL",
      "sql": "SELECT POOL_NAME,CURRENT_SIZE,DEFINED_SIZE,MAXIMUM_ACTIVE_THREADS,CURRENT_THREADS,RESERVED
    }
  ]
}
```

IBM i
continuous innovation
continuous integration

Data Visualization with Grafana

Grafana Overview

- Open-source analytics, visualization, and monitoring solution
- Driven by Grafana Labs. Plenty of open-source components. See https://grafana.com/
- Over 300 plugins available



Prometheus Visualization with Grafana

IBM i
continuous innovation
continuous integration

General / Prometheus Dashboard 2

Last 6 hours

Temporary storage

20500
20000
19500
19000

09:30 10:00 10:30 11:00 11:30 12:00 12:30 13:00 13:30 14:00 14:30 15:00

{__name__="ut25bp18__STATS__CURRENT_TEMPORARY_STORAGE", instance="9.5.10.206:9853", job="prometheus"}

Remote connections

8
6
4
2

09:30 10:00 10:30 11:00 11:30 12:00 12:30 13:00 13:30 14:00 14:30 15:00

{__name__="ut25bp18__REMOTE_CONNECTIONS", instance="9.5.10.206:9853", job="prometheus"}

ASP used (%)

86.0

Average CPU utilization

200
150
100
50
0

09:30 10:00 10:30 11:00 11:30 12:00 12:30 13:00 13:30 14:00 14:30 15:00

{__name__="ut25bp18__SYSACT__AVERAGE_CPU_UTILIZATION", instance="9.5.10.206:9853", job="prometheus"}

Active jobs in system

270
260
250
240

09:30 10:00 10:30 11:00 11:30 12:00 12:30 13:00 13:30 14:00 14:30 15:00

{__name__="ut25bp18__STATS__ACTIVE_JOBS_IN_SYSTEM", instance="9.5.10.206:9853", job="prometheus"}

Scrape duration

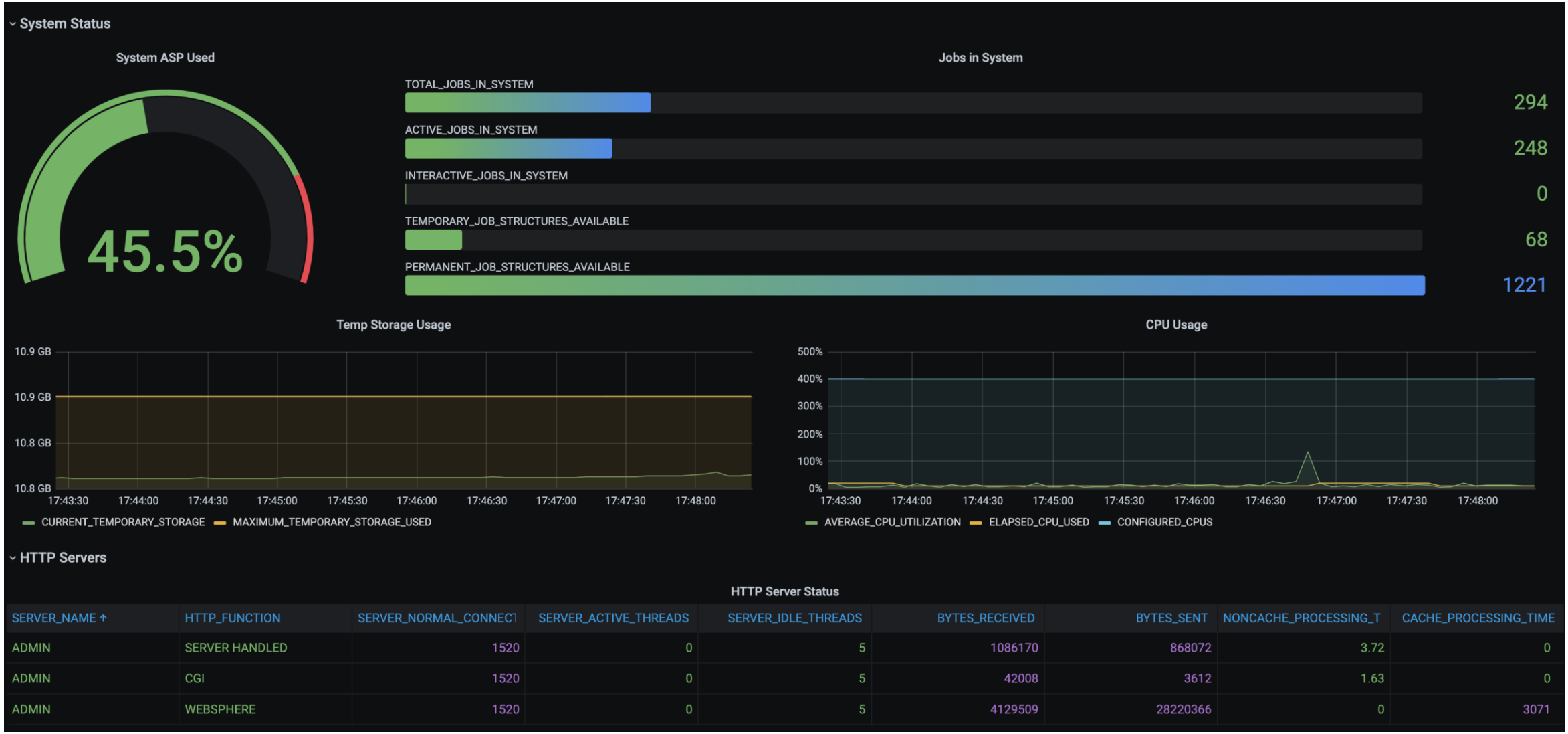
0.100
0.0800
0.0600
0.0400
0.0200
0

09:30 10:00 10:30 11:00 11:30 12:00 12:30 13:00 13:30 14:00 14:30 15:00

{__name__="scrape_duration_seconds", instance="9.5.10.206:9853", job="prometheus"}

Grafana Backend (Direct, No Prometheus)

IBM i
continuous innovation
continuous integration



SERVER_NAME ↑	HTTP_FUNCTION	SERVER_NORMAL_CONNECT	SERVER_ACTIVE_THREADS	SERVER_IDLE_THREADS	BYTES_RECEIVED	BYTES_SENT	NONCACHE_PROCESSING_T	CACHE_PROCESSING_TIME
ADMIN	SERVER HANDLED	1520	0	5	1086170	868072	3.72	0
ADMIN	CGI	1520	0	5	42008	3612	1.63	0
ADMIN	WEBSPHERE	1520	0	5	4129509	28220366	0	3071

Grafana with or without Prometheus

IBM i
continuous innovation
continuous integration

	Prometheus	Straight to Grafana
Persistent storage	Prometheus	Grafana
Persistent storage of unused metrics	Prometheus	--
Metric type	Numerics	Numerics/strings/other
Ecosystem	Extremely Broad	There
Scalability	Excellent	Good
IBM i requisites	None	Node.js
Initial setup	Easier	Easy

Event Monitoring with Manzan

Manzan Overview

IBM i
continuous innovation
continuous integration

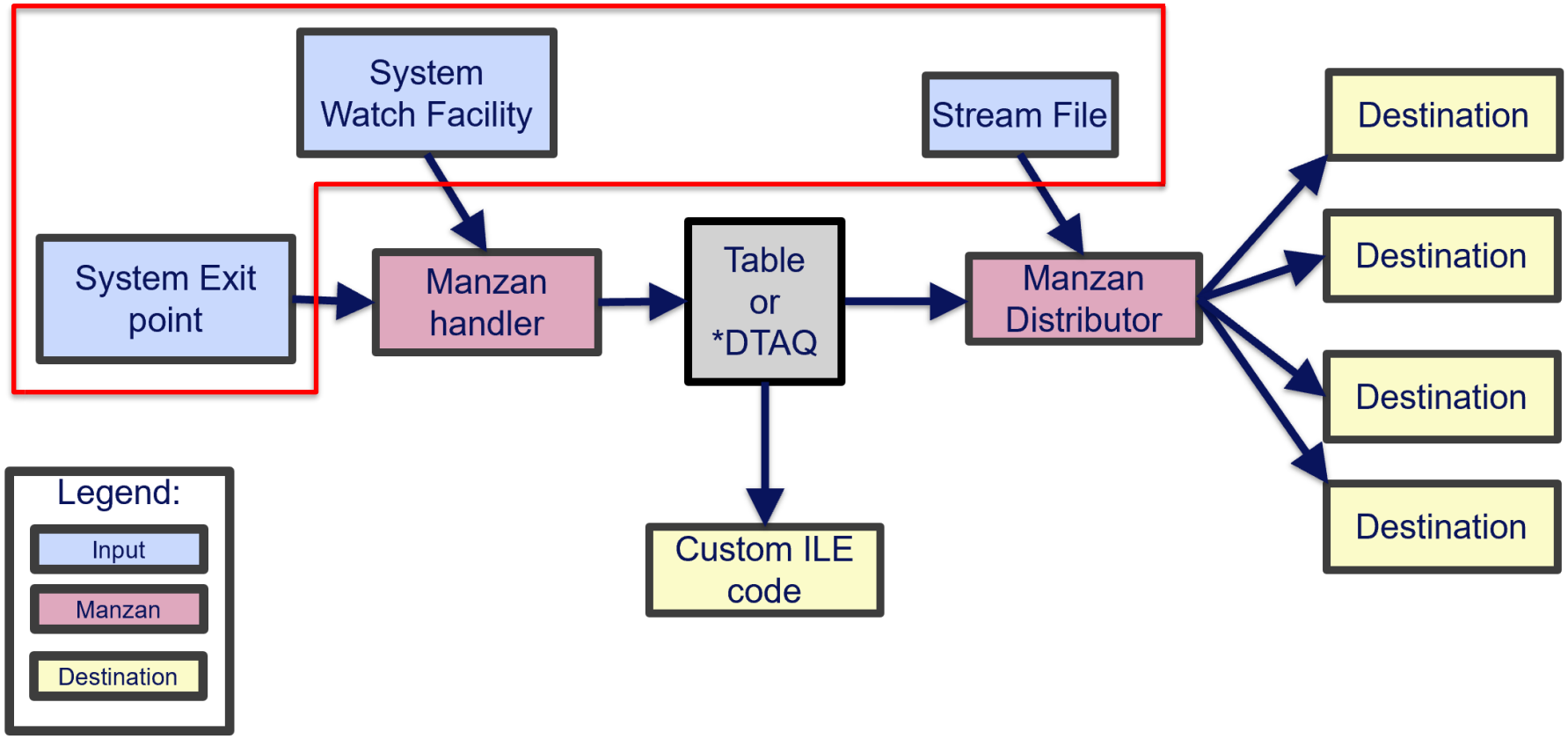
- Open-source event handling tool designed to simplify handling of system events
- Serves as a gateway for publishing IBM i events to a variety of endpoints:
  - User applications
  - External resources
  - Open-source technologies
- Example use cases:
  - Monitoring system events with a third-party open source or proprietary tool
  - More comprehensive integration with syslog facilities
  - Queryable system events
  - Consolidated auditing/reporting activity.

Understanding the Architecture: Inputs

IBM i
continuous innovation
continuous integration

Inputs: Sources of your data

- Stream file
- System watch facility (STRWCH)
  - MSGQ
  - LIC logs
  - PAL logs
- System exit points
- Audit journals (*coming soon*)



So practically what can Manzan monitor?

IBM i
continuous innovation
continuous integration

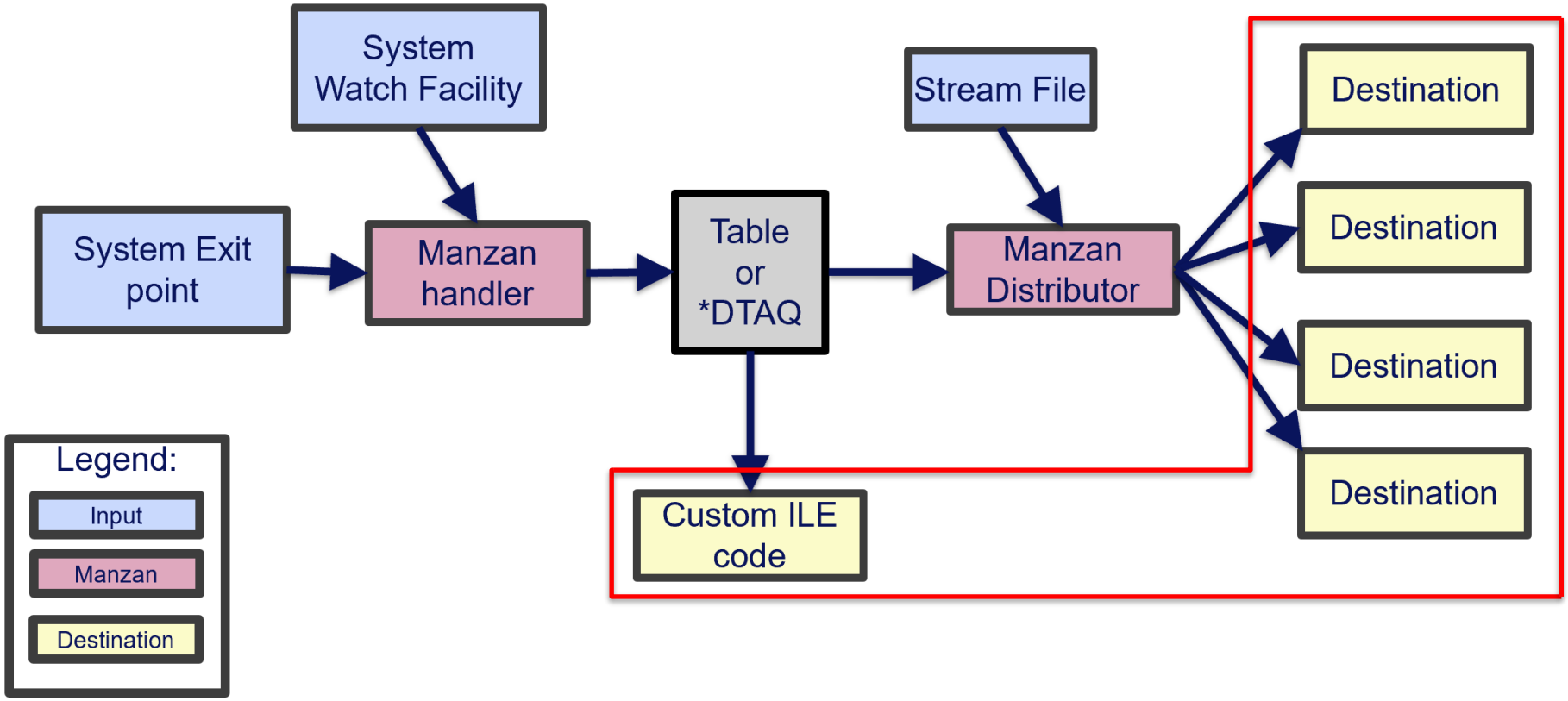
Application crashes	Log data	Specific entries in log data	System Limits alerts
History Log entries	Problem log entries	*SYSOPR messages	Specific job log messages
Audit journal events (future)	PAL entries	VLOGs	Fishy TCP connections (future)

Understanding the Architecture: Destinations

IBM i
continuous innovation
continuous integration

Destinations: Locations to send data

- Supported destinations:
  - HTTP/HTTPS endpoints (REST, etc)
  - Email (SMTP/SMTPS)
  - SMS (via Twilio)
  - Slack
  - FluentD
  - Kafka
  - Sentry
  - Grafana Loki
  - Google Pub/Sub
  - ActiveMQ
- Custom ILE Code



So can Manzan send data anywhere?

IBM i
continuous innovation
continuous integration

- Many desitnations are already working, but there are more to come
- Desired target not on the list? Please open an issue to the repository and let us know!
- Track supported destinations: https://theprez.github.io/Manzan/#/?id=where-can-i-send-these-events

- ActiveMQ ⏳
- AWS Simple Email Service (SES) ⏳
- AWS Simple Notification System (SNS) ⏳
- ElasticSearch ⏳
- Email (SMTP/SMTPS) ✅
- FluentD ✅
- Google Drive ⏳
- Google Pub/Sub ✅
- Grafana Loki ✅
- HTTP endpoints (REST, etc) ✅
- HTTPS endpoints (REST, etc) ✅
- Internet of Things (mqtt) ⏳
- Kafka ✅
- Mezmo ⏳
- Microsoft Teams ⏳
- PagerDuty ⏳
- Sentry ✅
- Slack ✅
- SMS (via Twilio) ✅
- Splunk ⏳

✅ = implemented 🌗 = partially implemented ⏳ = future

Configuring Inputs and Destinations

Configuration files are located in /QOpenSys/etc/manzan

app.ini

Used for general Manzan configuration
(you can leave the default contents)

```
[install]
library=MANZAN
```

data.ini

Used for configuring different data sources (inputs)

```
[<id>]
# where the data is coming from
type=<type>
# as defined in dests.ini
destinations=<destinations>

# other properties for <id> here..
```

dests.ini

Used for configuring different destinations

```
[<id>]
# the type of destination the data is going to
type=<type>

# other properties for <id> below (specific
# to each destination)...
```

Simple data.init configuration

IBM i
continuous innovation
continuous integration

Stream file

- Watch application log file (test.txt) and only take action when a line written to the file contains the string "error"

```
[logfile1]
type=file
file=test.txt
destinations=email_it, test_out
filter=error
format=$FILE_DATA$
```

System Watch

- Manage information from watch session with id "jesse"

```
[watcher1]
type=watch
id=jesse
destinations=test_out, slackme
enabled=false
```

Advanced data.init configuration

IBM i
continuous innovation
continuous integration

System Watch

- Manage information from watch session with id "jesse", which is configured to watch all messages in the IBM i history log
- Format the message into a human-sensible format
- Automatically start the watch when Manzan is started

```
[watchout]
type=watch
id=jesse
destinations=test_out, slackme
format=$MESSAGE_ID$ (severity $SEVERITY$): $MESSAGE$
strwch=WCHMSG((*ALL)) WCHMSGQ((*HSTLOG))
```

Example dests.ini configuration

IBM i
continuous innovation
continuous integration

Send an email

```
[email_b]
type=smtp
format=Hey, check out this information!! \n\n$FILE_DATA$
  server = my.smtpserver.com
  subject = Testemail
  from=me@mycompany.com
  to=me@mycompany.com
```

Send to Sentry

```
[sentry_out]
type=sentry
dsn=<slackdsn>
```

More example dests.ini configuration

IBM i
continuous innovation
continuous integration

Send SMS message with Twilio

```
[twilio_sms]
type=twilio
sid=[redacted]
token=[redacted]
to=+[redacted]
from=+[redacted]
```

Send to Slack

```
[slackme]
type=slack
channel=open-source-system-status
webhook=https://hooks.slack.com/services/TA3EF58G4...
```

Putting it together

IBM i
continuous innovation
continuous integration

dests.ini

```
[slackme]
type=slack
channel=open-source-system-status
webhook=https://hooks.slack.com/services/TA3EF58G4...
```

data.ini

```
[watchout]
type=watch
id=jesse
destinations=slackme
format=$MESSAGE_ID$ (severity $SEVERITY$): $MESSAGE$
strwch=WCHMSG((*ALL)) WCHMSGQ((*HSTLOG))
```

What does it look like in Slack?

IBM i
continuous innovation
continuous integration

open-source-system-status – Dec 20th, 2022

ibm i system monitor APP 9:01 PM

CPIAD09 (severity 0): User LINUX from client 9.163.40.192 connected to job 492988/QUSER/QZDASOINIT in subsystem QUSRWRK in QSYS on 12/20/22 20:17:06.

Future Enhancements to Manzan

IBM i
continuous innovation
continuous integration

- Trigger events based on audit journals

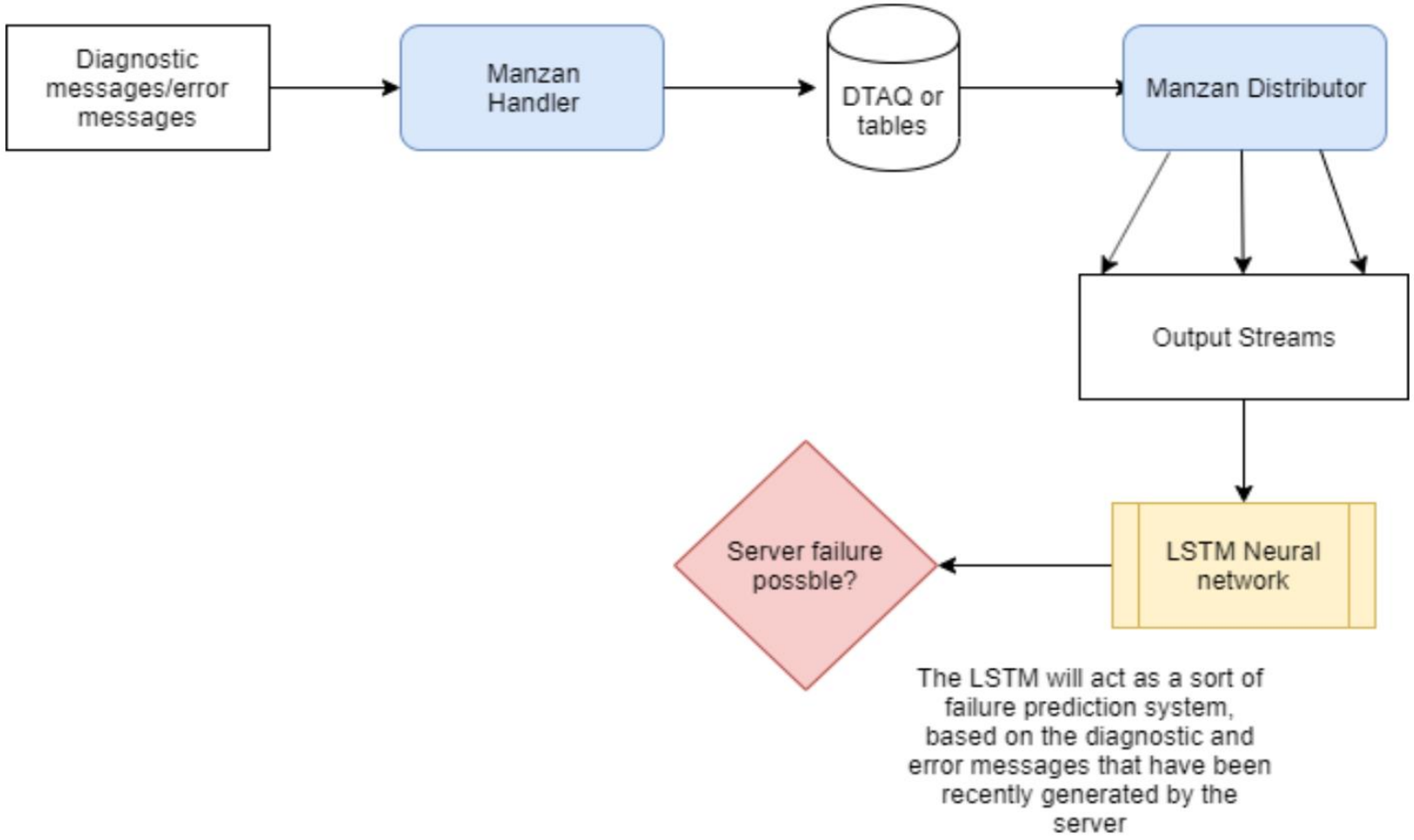
- Exit point processing (currently this is completely unimplemented)

- Metrics exported to Prometheus, for instance:
  - How many MCH exceptions have happened in production applications?
  - How many severity 40+ history log entries?
  - How many errors showing in web server logs?
  - How many PASE vlogs have been created?

- Ability to trigger events from Prometheus exporter, for instance:
  - Memory pool usage anomalies
  - Increased HTTP server traffic
  - Unexpected amount of remote connections
  - Operational data beyond thresholds

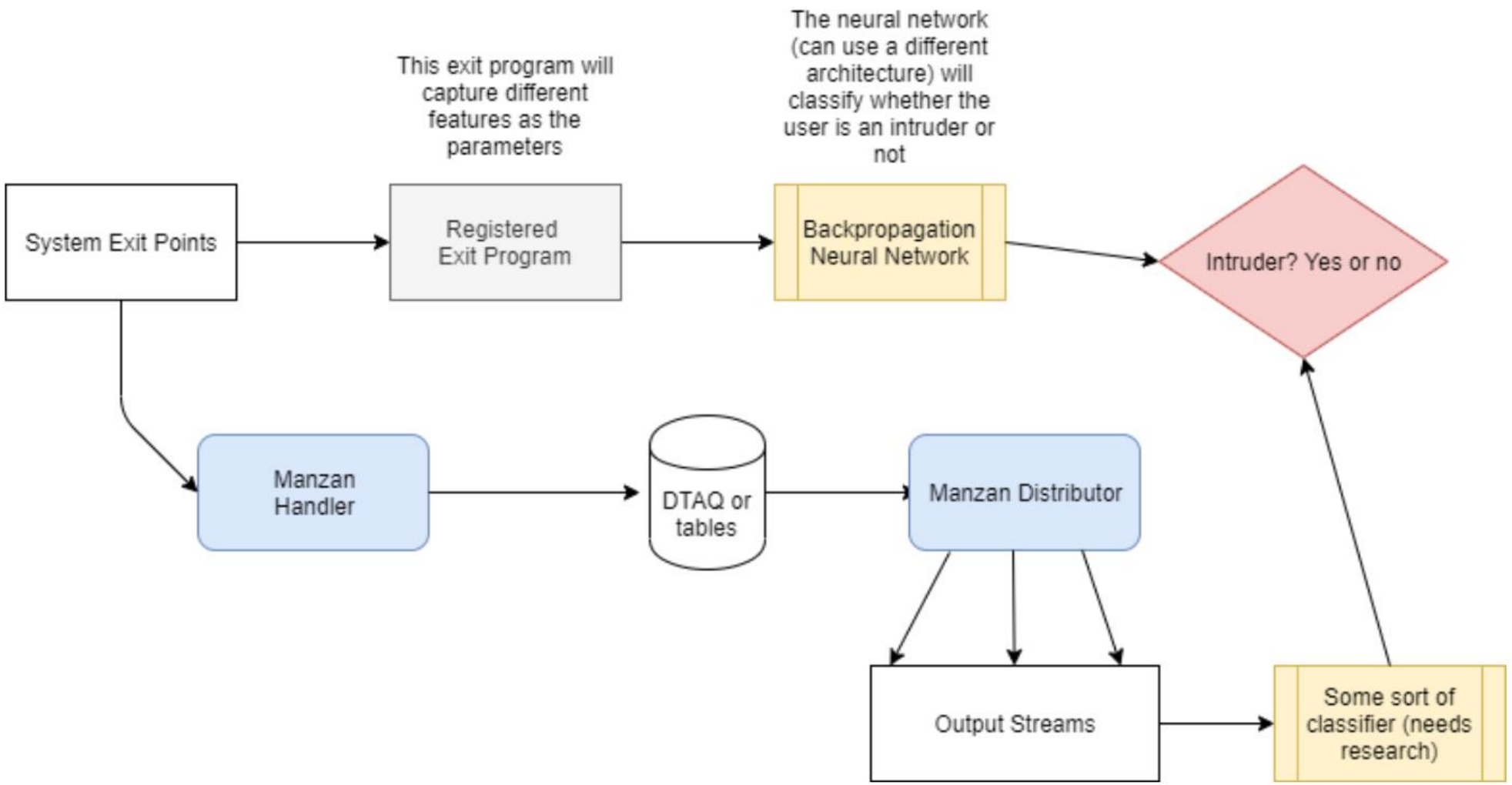
Manzan + AI

AI-Based System Monitoring: Failure Prediction System



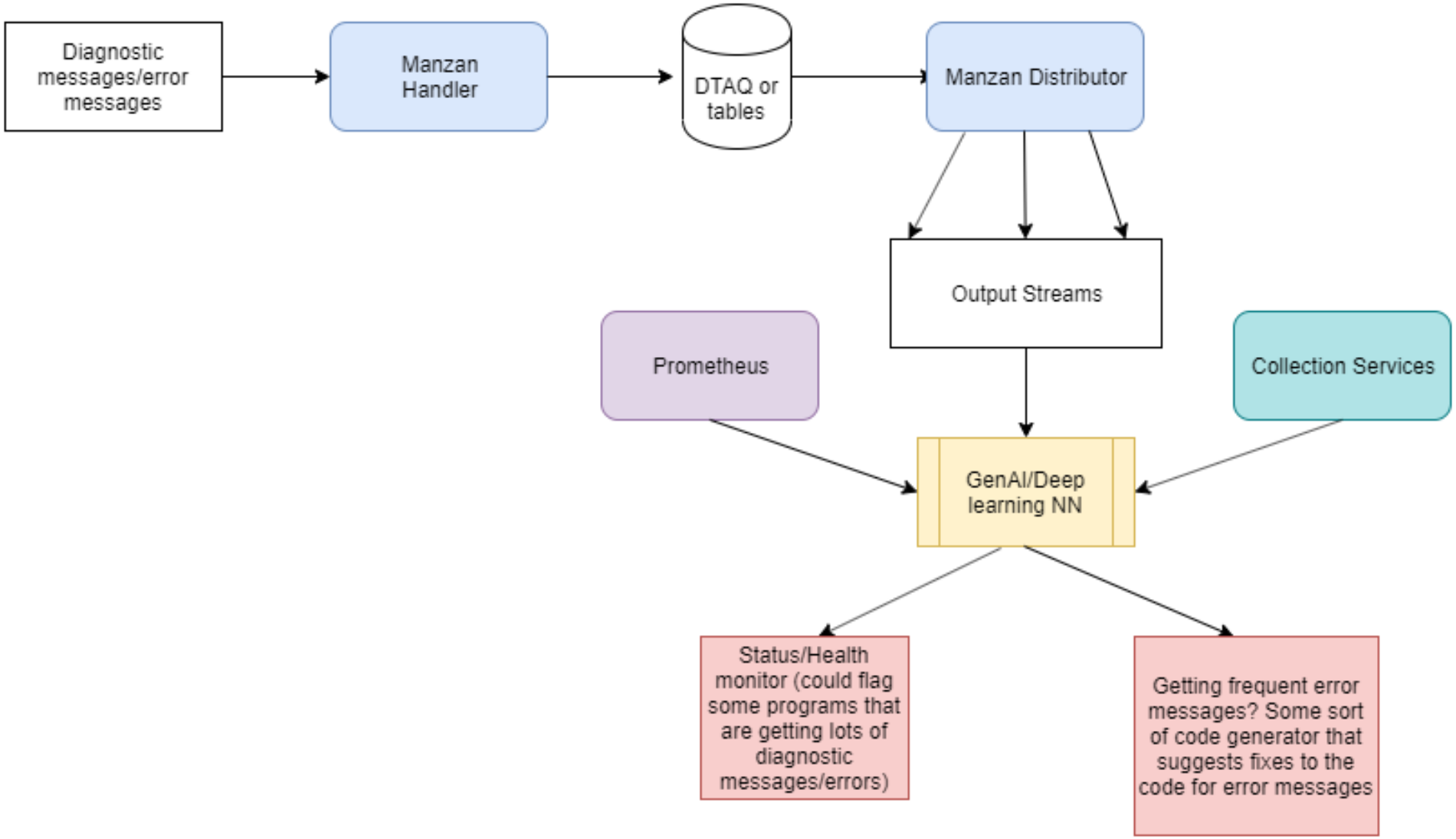
AI-Based System Monitoring: Intrusion Detection System

IBM i
continuous innovation
continuous integration



AI-Based System Monitoring: Health and Performance Assistant

IBM i
continuous innovation
continuous integration



Takeaways

IBM i
continuous innovation
continuous integration

01

Grafana is a must-have technology

02

Prometheus is a great way to enable ongoing monitoring

03

Manzan is a great way to handle various events and integrate with tons of solutions

Any Questions?

For More Information

IBM i
continuous innovation
continuous integration

Links You Need	Twitter	#Hashtags
IBM i Home Page: https://www.ibm.com/it-infrastructure/power/os/ibm-i (find link to Forrester Study and updated IBM i Strategy Whitepaper) IBM Strategy Whitepaper: https://www.ibm.com/it-infrastructure/us-en/resources/power/i-strategy-roadmap/ IBM Client Success: https://www.ibm.com/it-infrastructure/us-en/resources/power/ibm-i-customer-stories/ Support Life Cycle: https://www.ibm.com/support/lifecycle/ License Topics: https://www-01.ibm.com/support/docview.wss?uid=nas8N1022087 Fortra IBM i Marketplace Survey https://www.fortra.com/resources/guides/ibm-i-marketplace-survey-results	@IBMSystems @COMMONug @IBMChampions @IBMSystemsISVs @IBMiMag @ITJungleNews @SAPonIBMi @SiDforIBMi	#PowerSystems #IBMi #IBMAIX #POWER9 #LinuxonPower #OpenPOWER #HANAonPower #ITinfrastructure #OpenSource #HybridCloud #BigData