FROM CODE TO CLOUD:
THE DEVOPS EVOLUTION OF SMS SPAM DETECTION

Group 14: Boris Annink, Conall Lynch, Wilhelm Marcu, Jeffrey Meerovici Goryn, Alessandro Valmori

AGENDA

1. VERSIONING & RELEASE

2. PROVISIONING

3. DEPLOYMENT

4. RATE LIMITING & MONITORING

5. EXPERIMENTATION & EXTENSION

6. Q&A

VERSIONING & RELEASE

- Semantic versioning: The standard MAJOR.MINOR.PATCH structure is employed.

- Single Source of Truth (SSOT) for the front and backend are stored in:

  - pom.xml (app)

  - version.txt (model-service)

```
12      <groupId>sms</groupId>
13      <artifactId>frontend</artifactId>
14      <version>0.0.4</version>
```
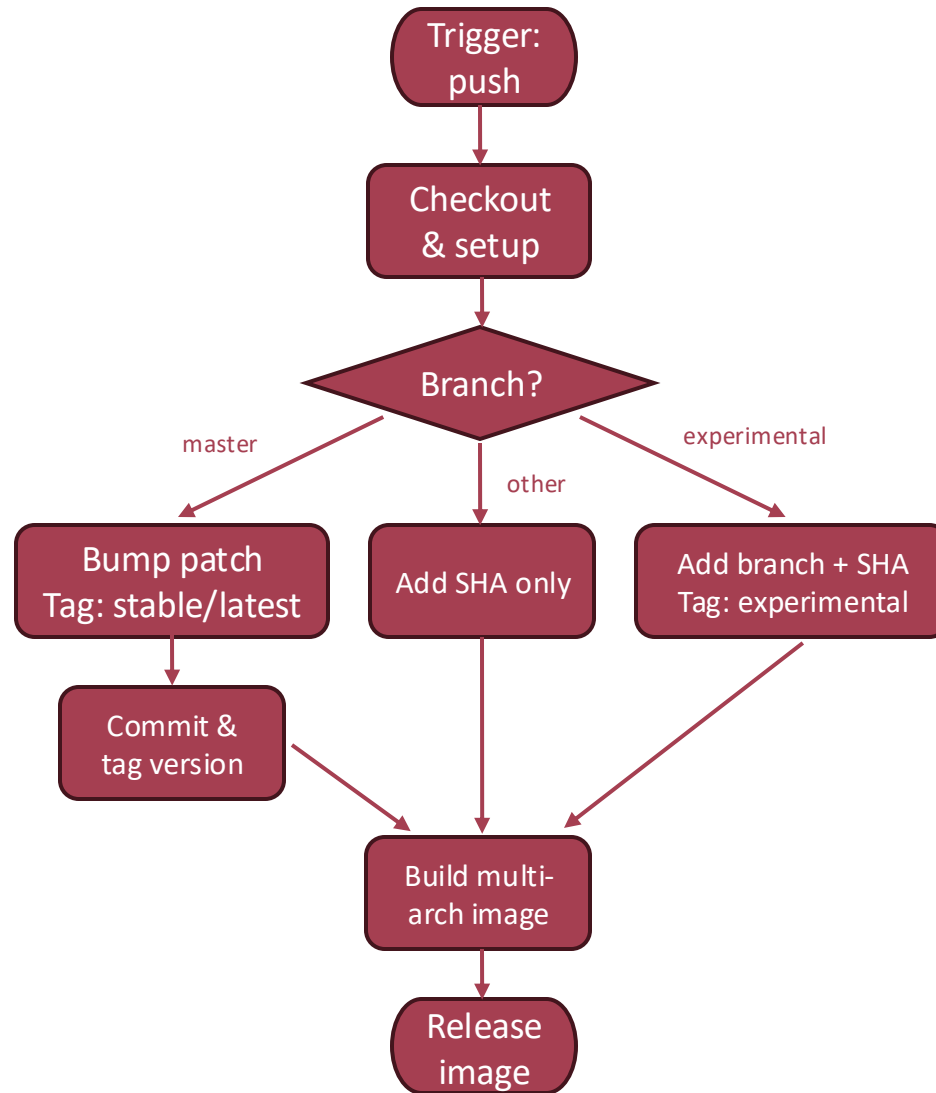
Versioning SSOT for Frontend

- Automated workflows via GitHub Actions:

  - **App:** Automated Releases

  - **Model-Service:** Automated Training Runs & Releases

- Triggered on:

  - Push to Master

  - PR Events

  - Manual Trigger for testing purposes

- Tags for:

  - Regular Release

  - Experimental Branch

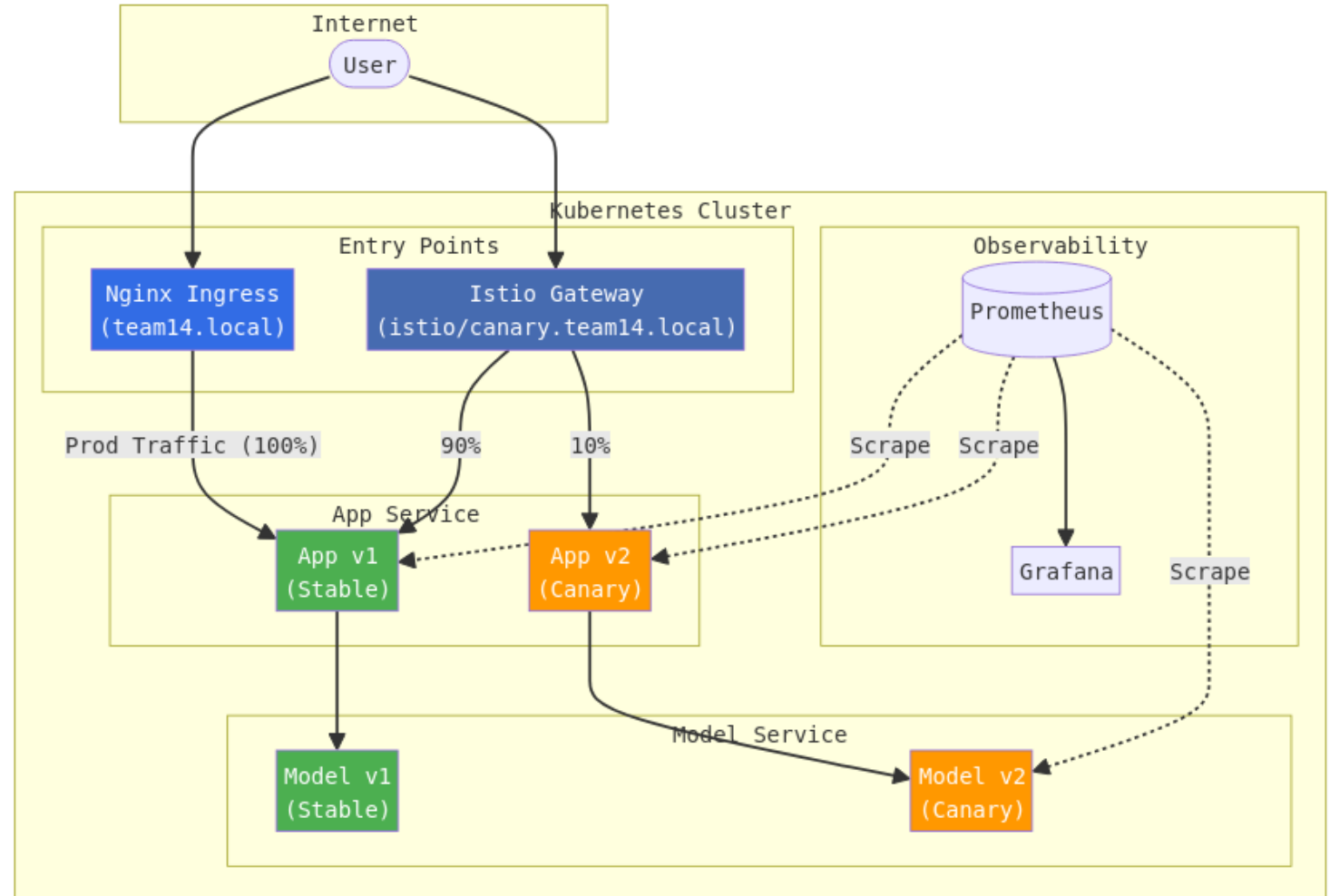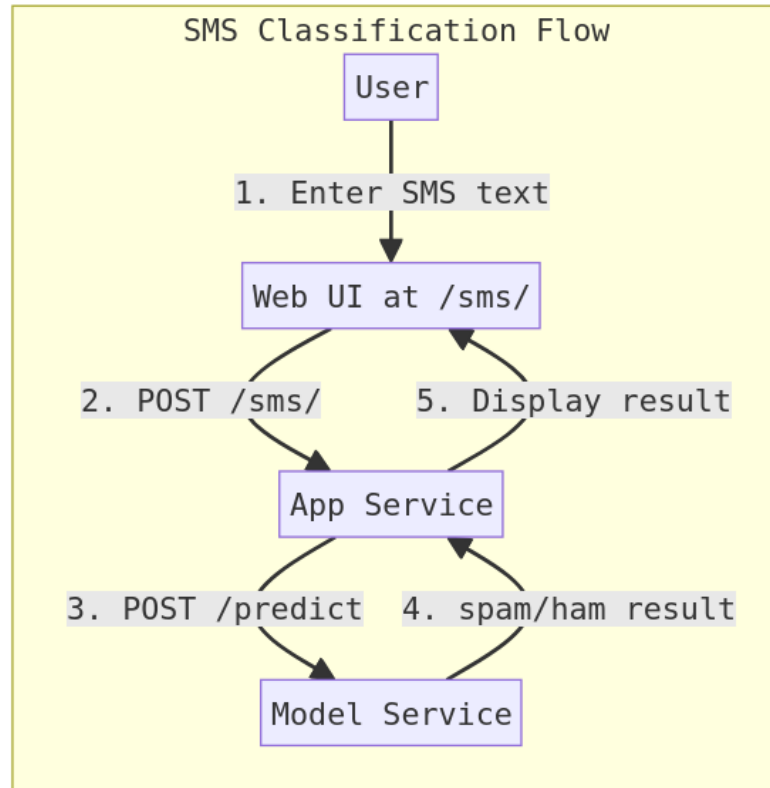*Frontend Release Workflow*

RELEASE PIPELINE

PROVISIONING

- Ansible Playbooks are employed to provision the cluster.

- Two Types: Ctrl and Node

  - **Node:** Runs Standard Setup playbooks and a **node** specific playbook which joins the node to the cluster.

  - **Ctrl:** Our control node. Runs standard playbooks and **ctrl** specific playbook which installed the ctrl requirements, creates a shared folder at /mnt/shared/ and generates the join command

- Drawbacks discussed in the *Extension* section

- Our implementation runs on a Vagrant Virtual Machine Cluster.

- Implementation is cluster agnostic.

- Bash scripts provided for easier provisioning with:

  - Minikube (local testing)

  - Docker

  - Vagrant VMs

DEPLOYMENT

SMS Classification Flow

User
↓ 1. Enter SMS text
Web UI at /sms/
↓ 2. POST /sms/ ↑ 5. Display result
App Service
↓ 3. POST /predict ↑ 4. spam/ham result
Model Service

Internet
User

Kubernetes Cluster

Entry Points
Nginx Ingress (team14.local)
Istio Gateway (istio/canary.team14.local)

Prod Traffic (100%) — 90% — 10%

App Service
App v1 (Stable)
App v2 (Canary)

Model Service
Model v1 (Stable)
Model v2 (Canary)

Observability
Prometheus
Scrape — Scrape — Scrape
Grafana

TRAFFIC ROUTING

- Ingress provides access to stable release

- Istio service mech for advanced routing

  - Access both stable and experimental

  - Destination rule for configuring split

  - 90/10

  - Cookies for sticky sessions
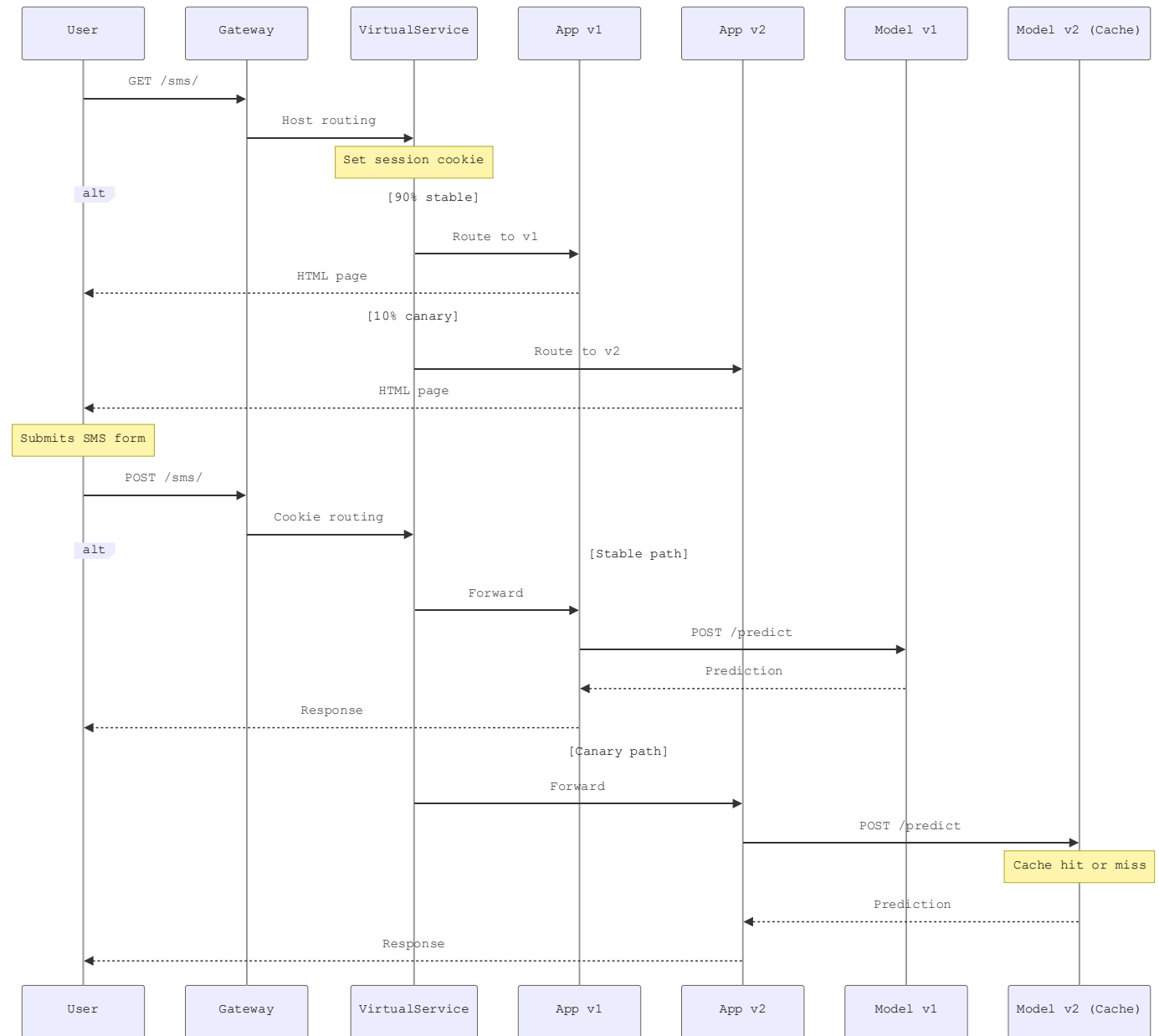
RATE LIMITING

- Need a way to deal with excessive traffic

  - Malicious: Denial of Service attacks

  - Unintentional: bugged client in endless loop

- Service degradation

- Resource starvation

- Increased costs

- Implemented at the Istio service mesh level

- EnvoyFilter to extract descriptors

- 10 req/min on /sms/ path

```
jmeerovici@jmeerovici-HP-ZBook-Power-15-6-inch-G9-Mobile-Workstation-PC:~/operation$ for i in {1..15}; do
 printf "%s " "$(date '+%Y-%m-%d %H:%M:%S')"
 curl -s -o /dev/null -w "%{http_code}\n" \
   -H "Host: istio.team14.local" \
   -H "X-User-Id: Bob" \
   http://$(minikube ip):$(kubectl get svc -n istio-system istio-ingressgateway \
     -o jsonpath='{.spec.ports[?(@.port==80)].nodePort}')/sms/
 sleep 1
done
2026-01-24 21:35:10 200
2026-01-24 21:35:11 200
2026-01-24 21:35:13 200
2026-01-24 21:35:14 200
2026-01-24 21:35:15 200
2026-01-24 21:35:17 200
2026-01-24 21:35:18 200
2026-01-24 21:35:19 200
2026-01-24 21:35:21 200
2026-01-24 21:35:22 200
2026-01-24 21:35:23 429
2026-01-24 21:35:25 429
2026-01-24 21:35:26 429
2026-01-24 21:35:27 429
2026-01-24 21:35:29 429
```
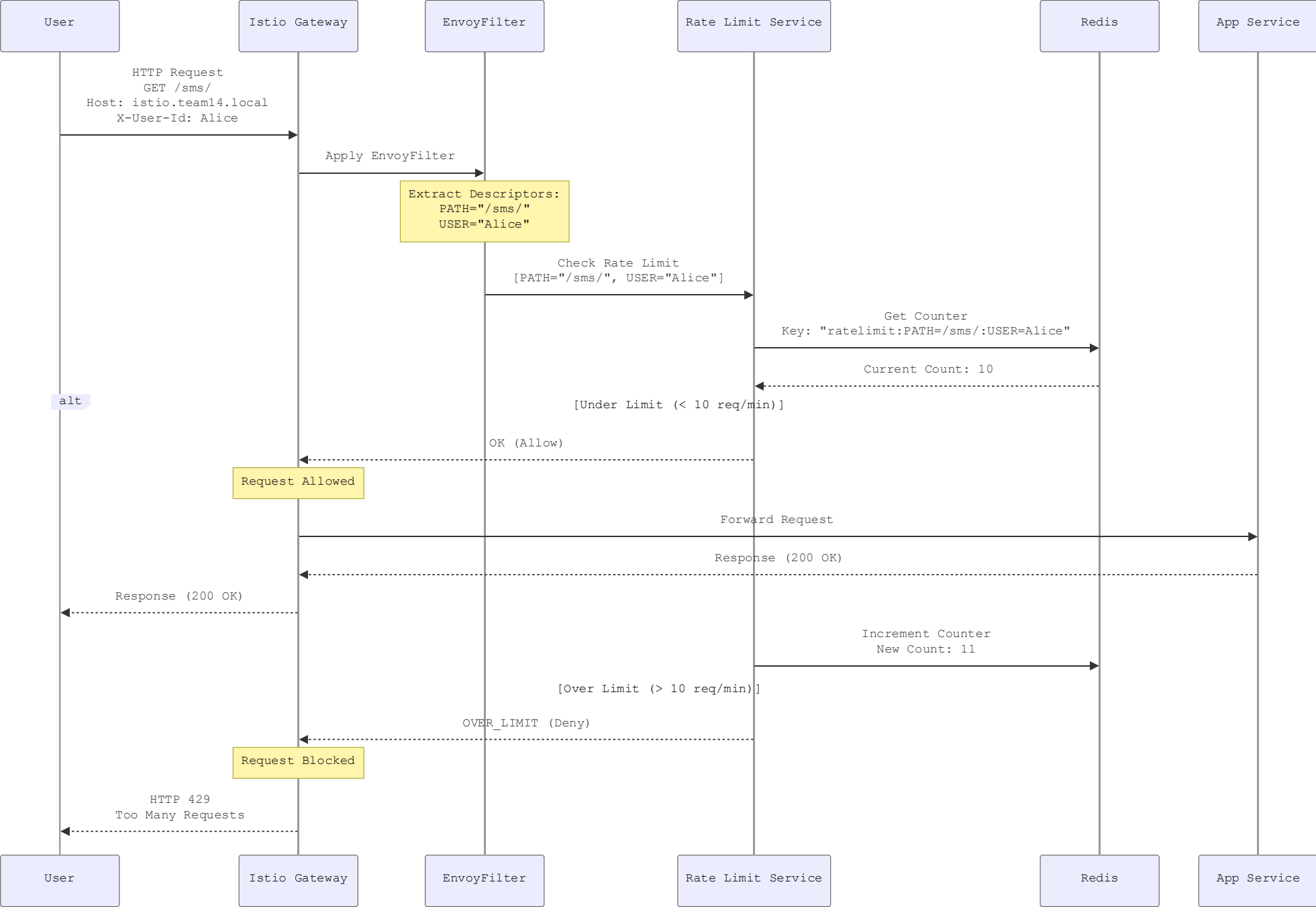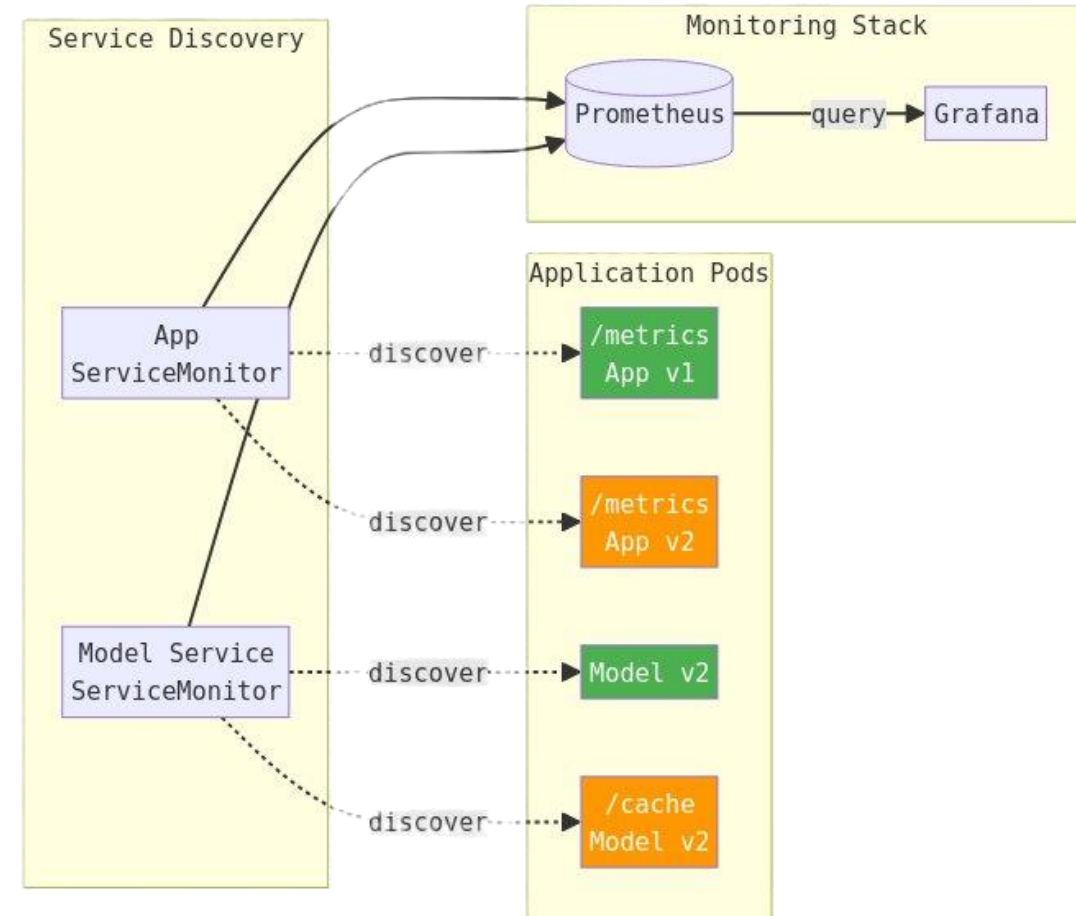
METRICS

- /metrics endpoint exposes data

  - % messages classified as ham

  - # active users

  - request rate

  - average latency

  - CPU / memory usage

- ServiceMonitors periodically scrape endpoints

EXPERIMENTATION

- Currently: every request submitted to ML model

  - High overhead, repeated requests, scales poorly

- Idea: cache requests, only use ML model if needed

  - Size: 1000, TTL: 1h, FIFO eviction

  - Key = SHA-256 of message

- Caching feature can be toggled

  - Only used if X-Cache-Enabled header is set

- Metrics: prediction latency (ms), CPU usage (%)

- Hypothesis: reduce metrics by at least 20%
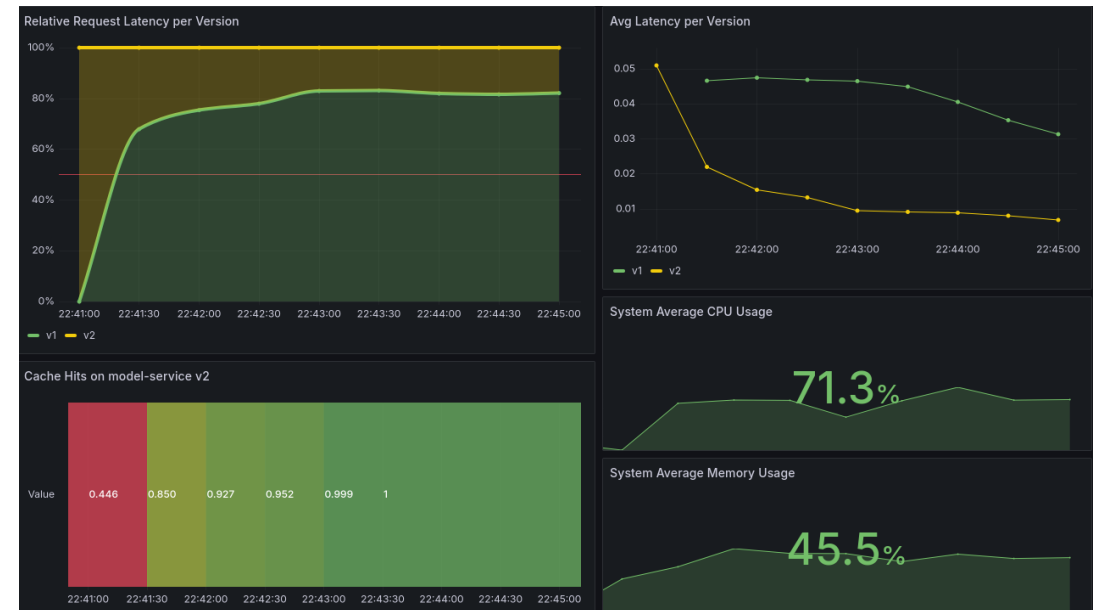
EXTENSION

- IaC used: Kubernetes, Helm, GH workflows, Vagrant, Ansible

- Inconsistency: Ansible is imperative, everything else is declarative

- Pain point: updating playbooks slow and tedious

  - Long provisioning times

  - Non-atomic updates: error = broken VM

  - No rollback mechanism

  - Not truly reproducible: convergent but not congruent

- Solution: NixOS: Linux distro built on the Nix package manager

  - Purely functional, declarative system config

  - Strong guarantees about correctness and reproducibility

  - Atomic updates: error = stay on prior version

  - Rollback built in

- Potential drawbacks: steep learning curve, packages must be specifically designed for Nix
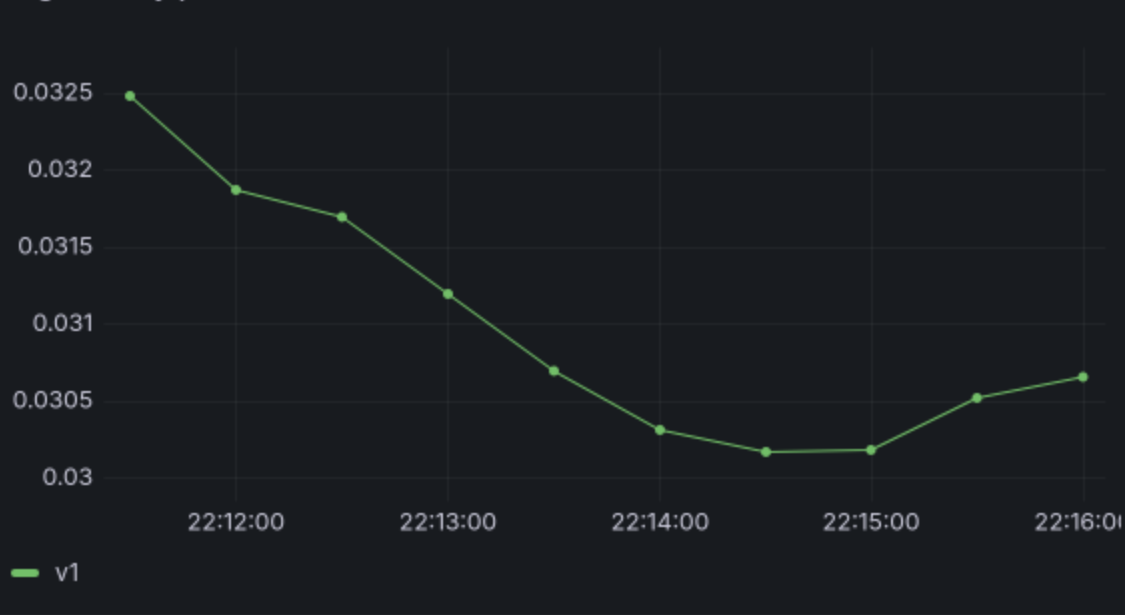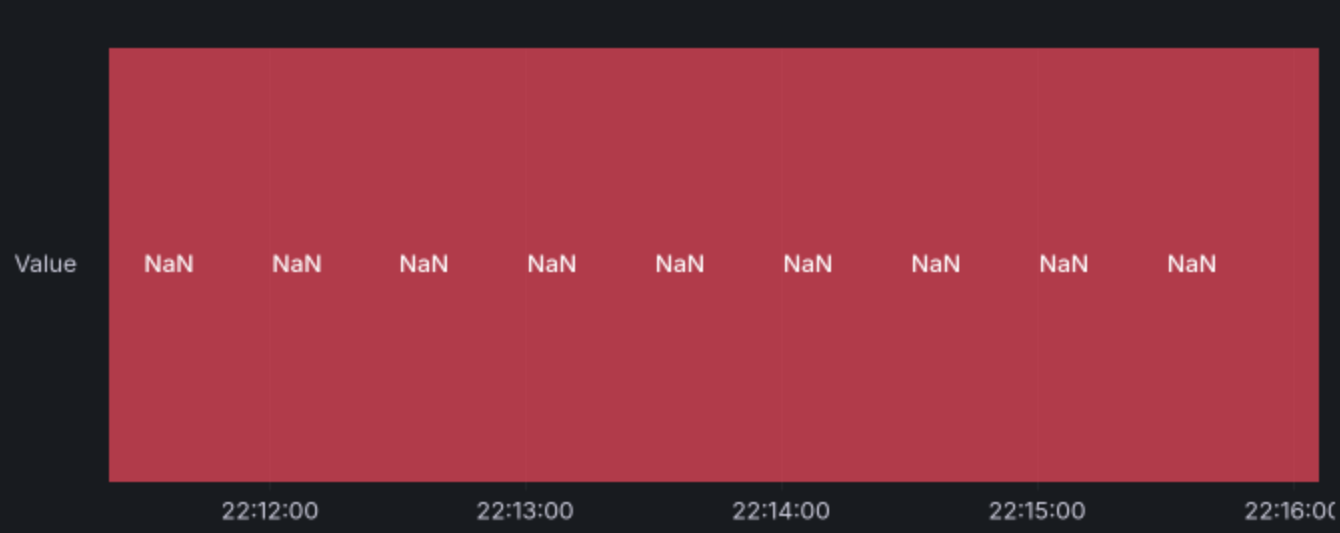
THANK YOU!

ANY QUESTIONS?

Relative Request Latency per Version

- v1

Avg Latency per Version

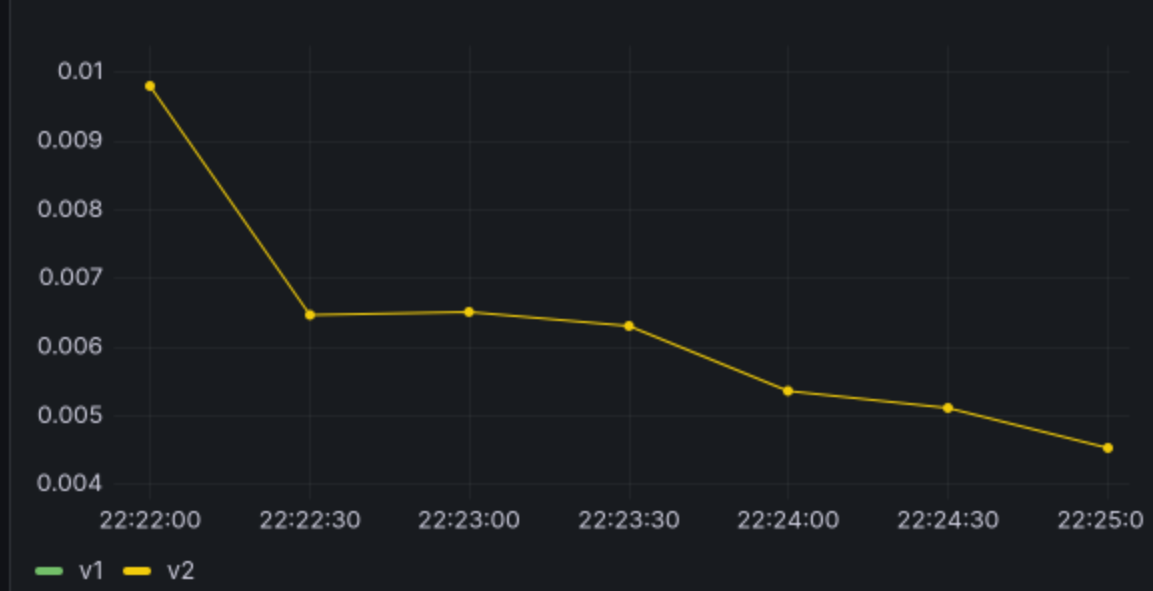

- v1

Cache Hits on model-service v2

Value | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN

System Average CPU Usage

62.7%

System Average Memory Usage

44.9%

Relative Request Latency per Version

100%
80%
60%
40%
20%
0%

22:22:00 22:22:30 22:23:00 22:23:30 22:24:00 22:24:30 22:25:00

— v1 — v2

Avg Latency per Version

0.01
0.009
0.008
0.007
0.006
0.005
0.004

22:22:00 22:22:30 22:23:00 22:23:30 22:24:00 22:24:30 22:25:0

— v1 — v2

Cache Hits on model-service v2

Value 0.738 0.923 0.951 0.967 1

22:22:00 22:22:30 22:23:00 22:23:30 22:24:00 22:24:30 22:25:00

System Average CPU Usage

9.75%

System Average Memory Usage

46.5%

Relative Request Latency per Version

- v1
- v2

Avg Latency per Version

- v1 - v2

Cache Hits on model-service v2

| Value | 0.446 | 0.850 | 0.927 | 0.952 | 0.999 | 1 |

System Average CPU Usage

71.3%

System Average Memory Usage

45.5%
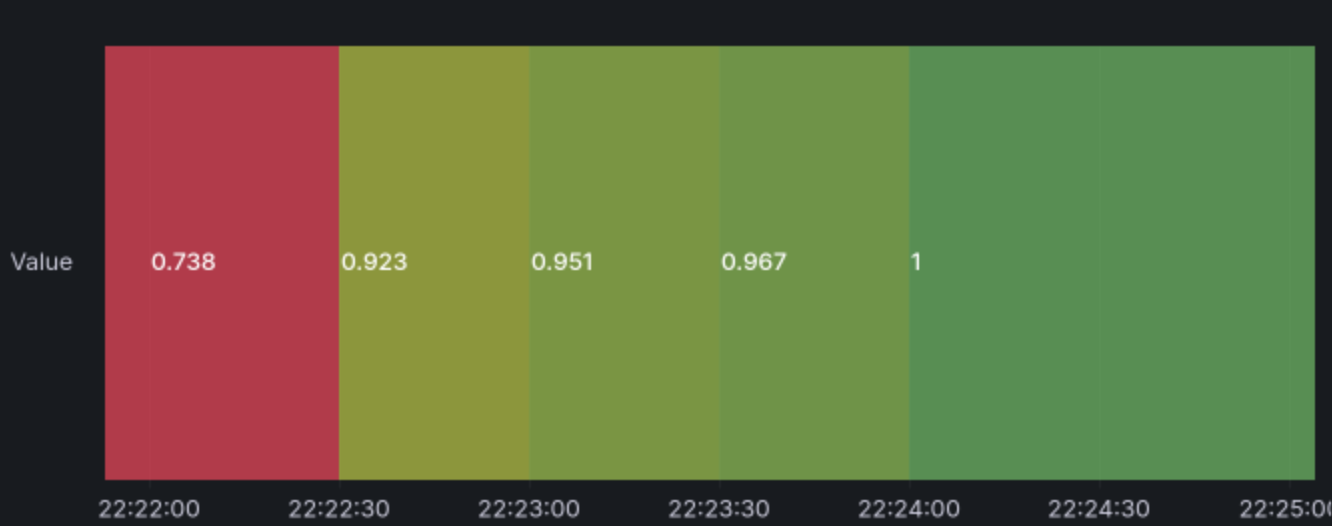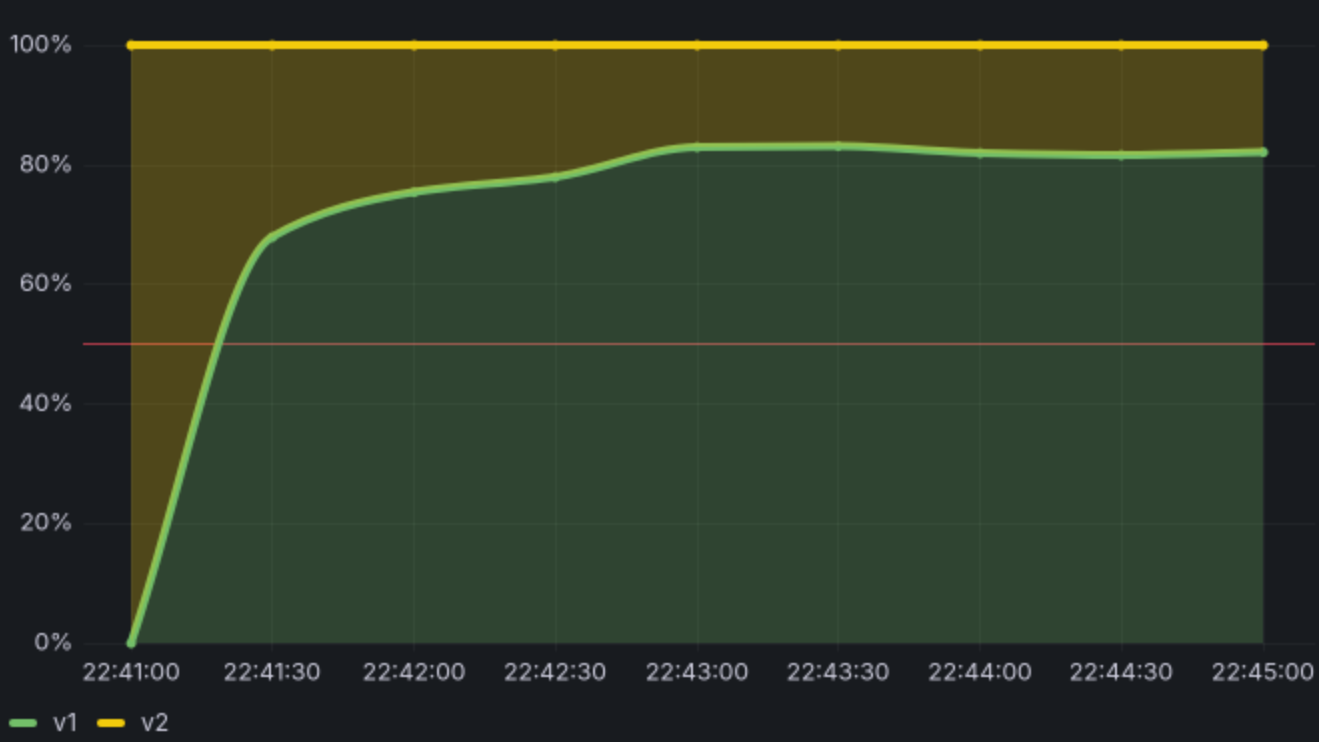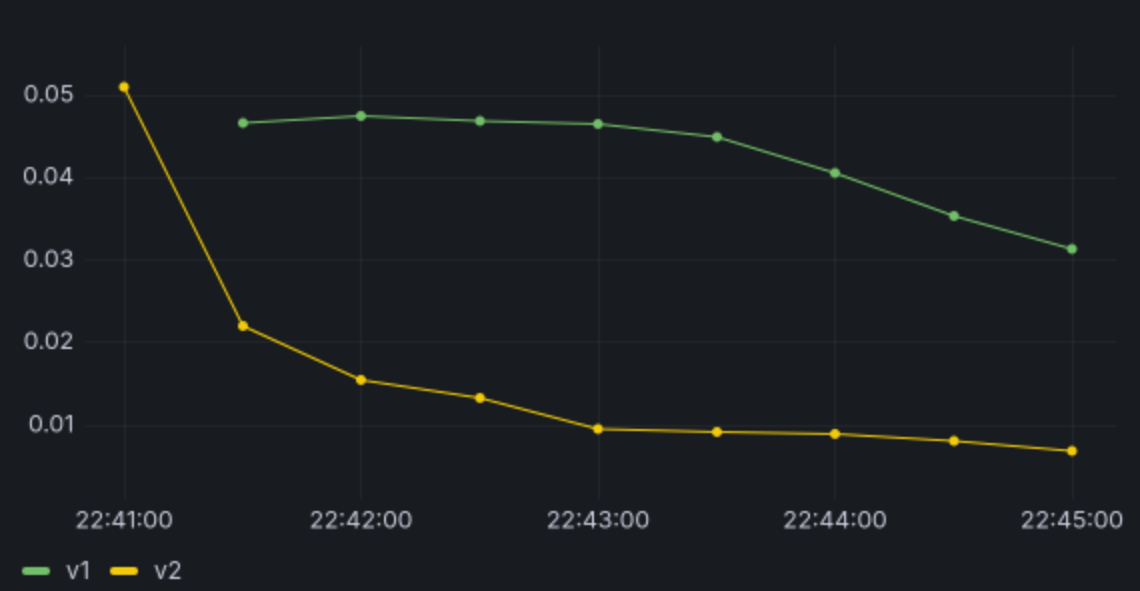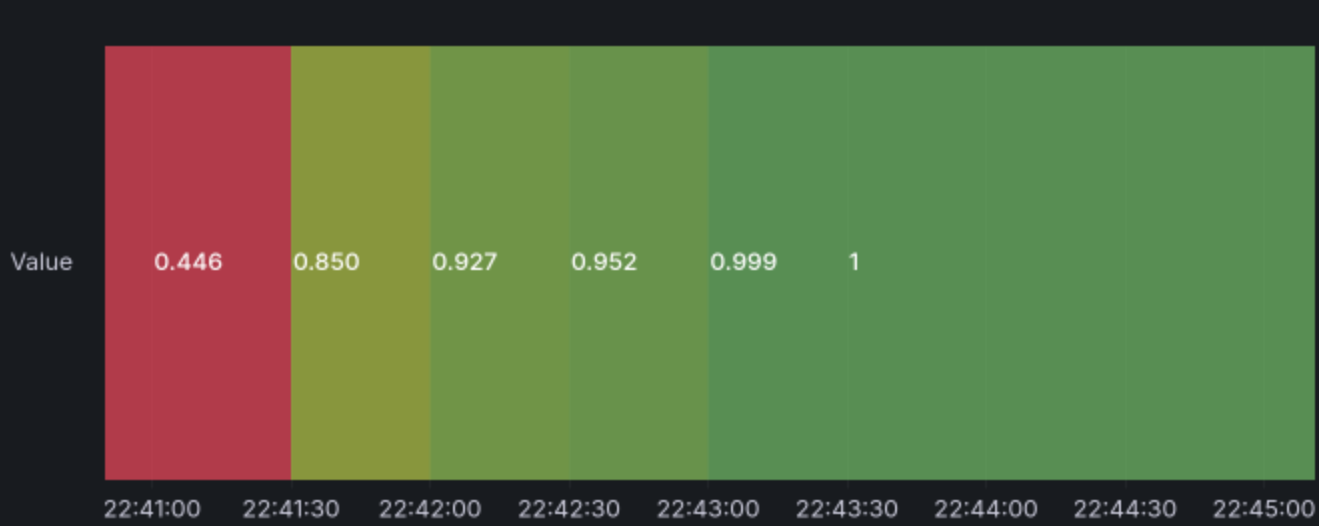
THE FOLLOWING SLIDES ARE PART OF THE TEMPLATE

PLEASE LEAVE THEM UNCHANGED (COPY THEM IF NEEDED)

FROM CODE TO CLOUD:
THE DEVOPS EVOLUTION OF SMS SPAM DETECTION

Group 14: Boris Annink, Conall Lynch, Wilhelm Marcu, Jeffrey Meerovici Goryn, Alessandro Valmori

AGENDA

1. INTRODUCTION

2. BUILDING CONFIDENCE

3. ENGAGING THE AUDIENCE

4. VISUAL AIDS

5. FINAL TIPS & TAKEAWAYS

6.

7. Q&A

THE POWER OF COMMUNICATION

OVERCOMING NERVOUSNESS

CONFIDENCE-BUILDING STRATEGIES

ENGAGING THE AUDIENCE

- Make eye contact with your audience to create a sense of intimacy and involvement

- Weave relatable stories into your presentation using narratives that make your message memorable and impactful

- Encourage questions and provide thoughtful responses to enhance audience participation

- Use live polls or surveys to gather audience opinions, promoting engagement and making sure the audience feel involved

SELECTING VISUAL AIDS

ENHANCING YOUR PRESENTATION

EFFECTIVE DELIVERY TECHNIQUES

This is a powerful tool in public speaking. It involves varying pitch, tone, and volume to convey emotion, emphasize points, and maintain interest:

- Pitch variation

- Tone inflection

- Volume control

Effective body language enhances your message, making it more impactful and memorable:

- Meaningful eye contact

- Purposeful gestures

- Maintain good posture

- Control your expressions

NAVIGATING Q&A SESSIONS

1. Know your material in advance

2. Anticipate common questions

3. Rehearse your responses

Maintaining composure during the Q&A session is essential for projecting confidence and authority. Consider the following tips for staying composed:

- Stay calm

- Actively listen

- Pause and reflect

- Maintain eye contact

SPEAKING IMPACT

Your ability to communicate effectively will leave a lasting impact on your audience

Effectively communicating involves not only delivering a message but also resonating with the experiences, values, and emotions of those listening

DYNAMIC DELIVERY

Learn to infuse energy into your delivery to leave a lasting impression

One of the goals of effective communication is to motivate your audience

METRIC	MEASUREMENT	TARGET	ACTUAL
Audience attendance	# of attendees	150	120
Engagement duration	Minutes	60	75
Q&A interaction	# of questions	10	15
Positive feedback	Percentage (%)	90	95
Rate of information retention	Percentage (%)	80	85

FINAL TIPS & TAKEAWAYS

Consistent rehearsal

- Strengthen your familiarity

Refine delivery style

- Pacing, tone, and emphasis

Timing and transitions

- Aim for seamless, professional delivery

Practice audience

- Enlist colleagues to listen & provide feedback

Seek feedback

Reflect on performance

Explore new techniques

Set personal goals

Iterate and adapt

SPEAKING ENGAGEMENT METRICS

IMPACT FACTOR	MEASUREMENT	TARGET	ACHIEVED
Audience interaction	Percentage (%)	85	88
Knowledge retention	Percentage (%)	75	80
Post-presentation surveys	Average rating	4.2	4.5
Referral rate	Percentage (%)	10	12
Collaboration opportunities	# of opportunities	8	10

THANK YOU

Brita Tamm

502-555-0152

brita@firstupconsultants.com

www.firstupconsultants.com