



Mastering User Journeys with ThousandEyes Transaction Testing

Elevate Your Digital Experience

Aditya Chellam
Customer Delivery Engineering Technical Leader
BRKCLD-1032

CISCO *Live!*



Webex App

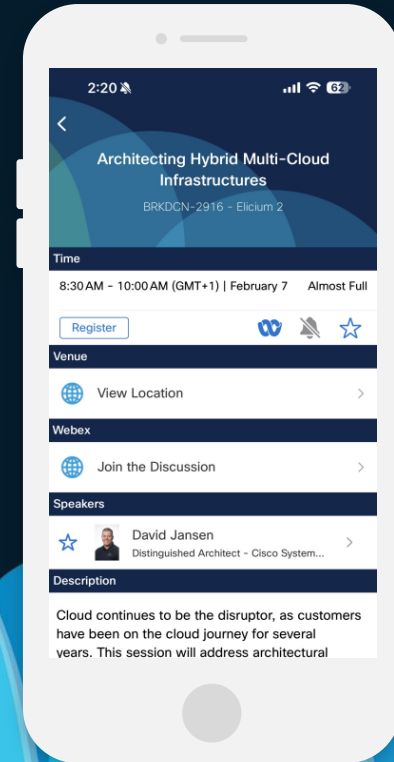
Questions?

Use the Webex app to chat with the speaker after the session

How

- 1 Find this session in the Cisco Events mobile app
- 2 Click “Join the Discussion”
- 3 Install the Webex app or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 28, 2025.

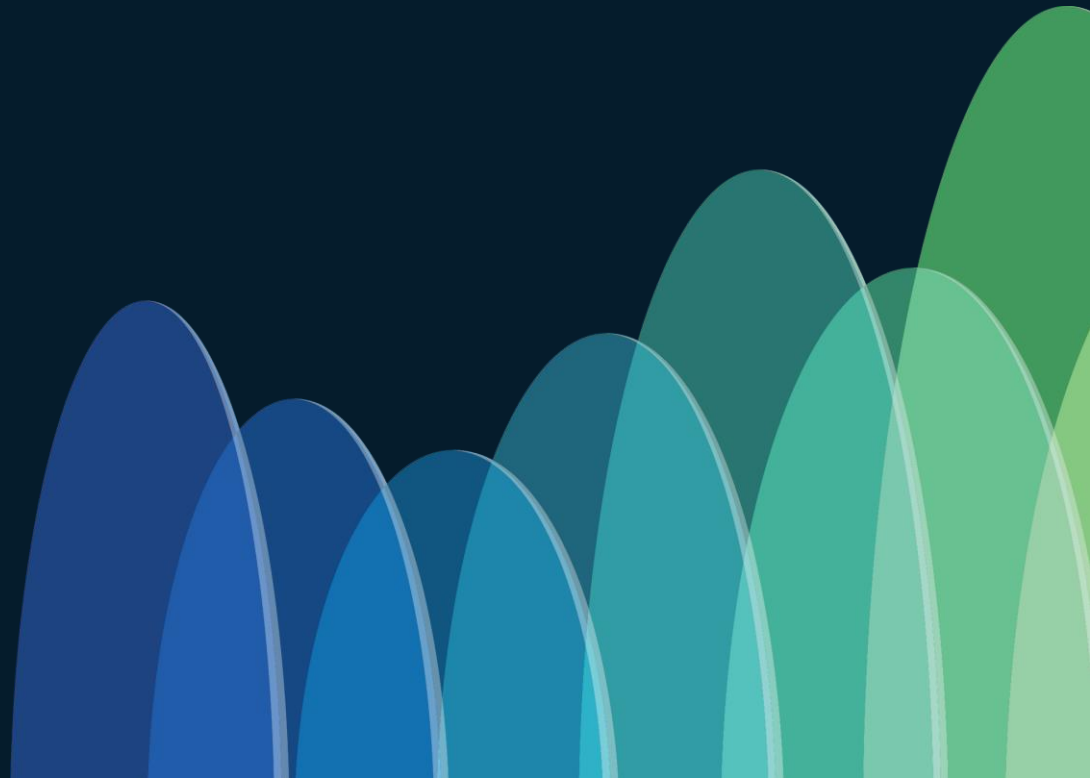




Agenda

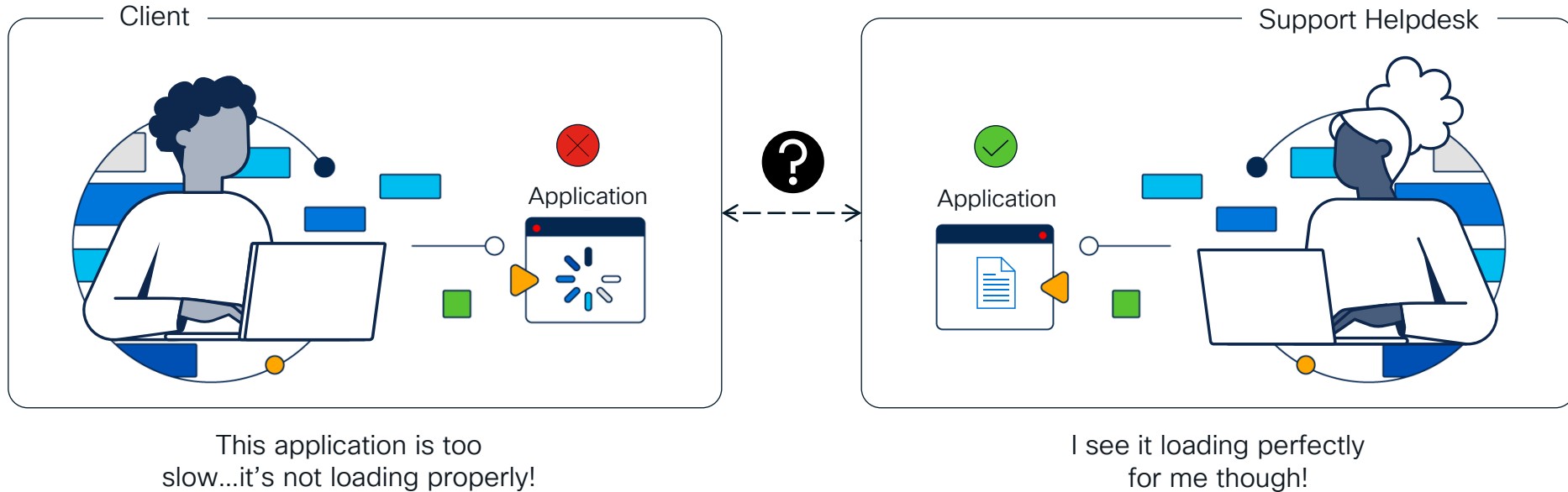
- Why Observability matters
- What is ThousandEyes
- Elevating customer digital experience with Assurance
- Transaction testing for customer journeys
- Proactive monitoring in action

Why Observability Matters



Why Observability Matters?

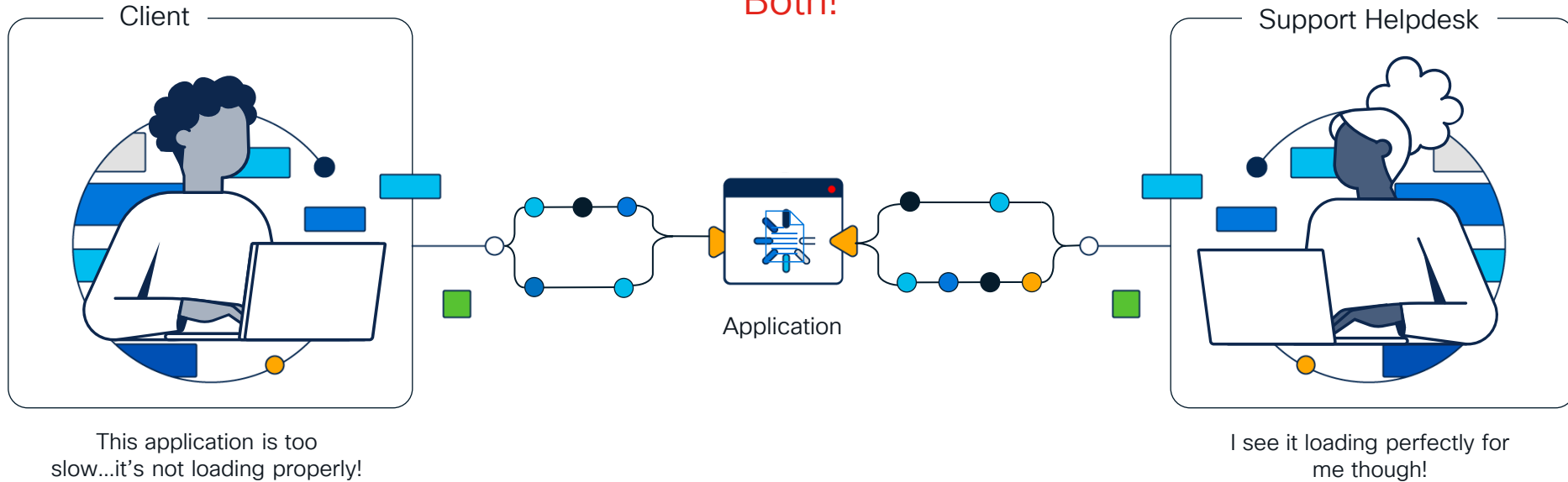
Who's Telling the Truth?



Why Observability Matters?

Who's Telling the Truth?

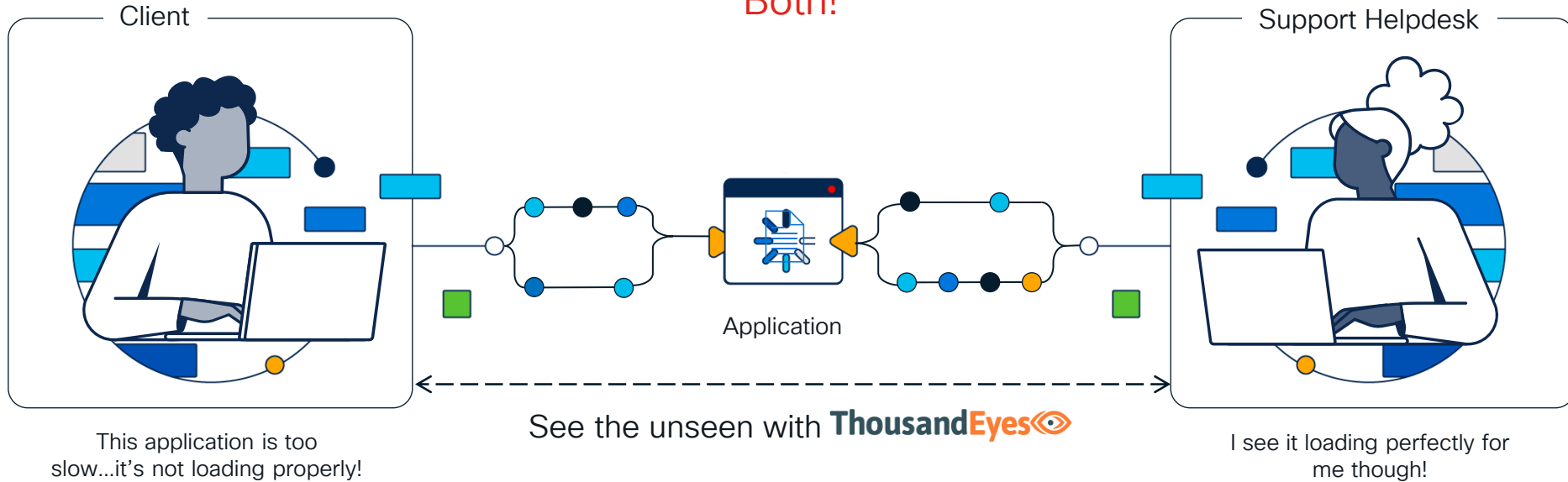
Both!



Why Observability Matters?

Who's Telling the Truth?

Both!



Why Observability Matters?

Regulatory requirements in different part of the world **require** observability for end-to-end **Network Assurance**

EMEA

Digital Operational
Resilience Act
(DORA)

AMER

SR 21-3 & Principles
for Operational
Resilience
(US Federal Reserve)

APJC

CPS 230 Operational
Risk Management
(APRA)

Why Observability Matters?

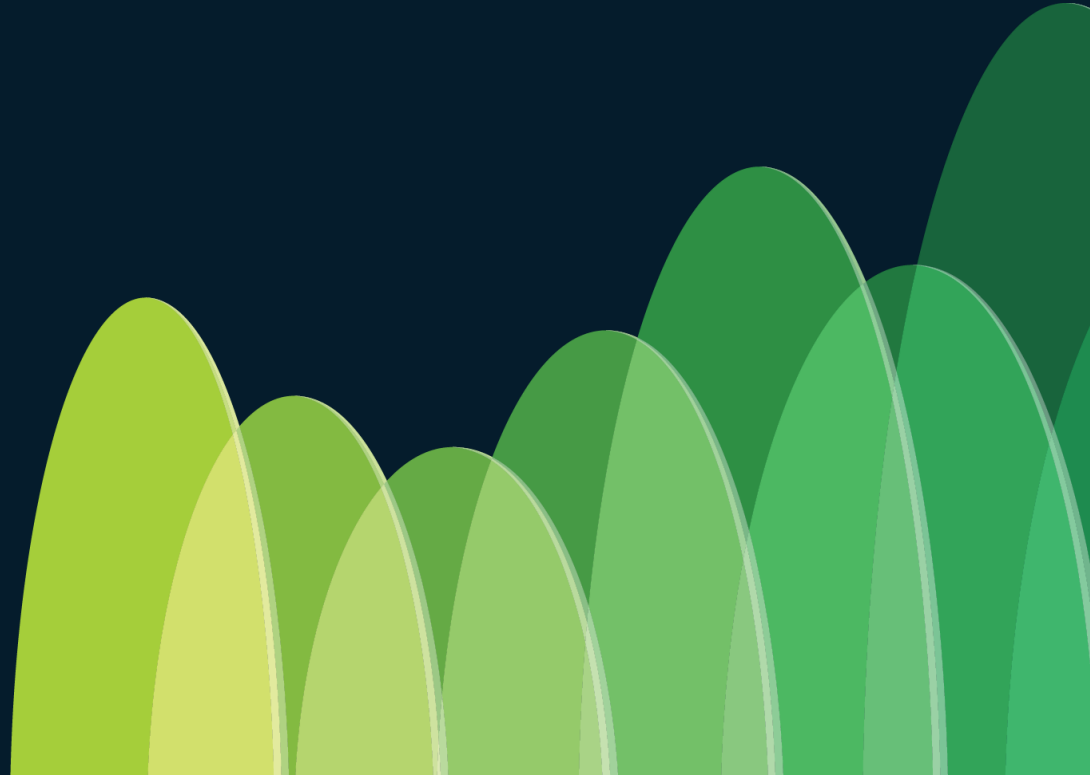
Regulatory requirements in different part of the world **require** observability for end-to-end **Network Assurance**

EMEA

Digital Operational Resilience Act (DORA)

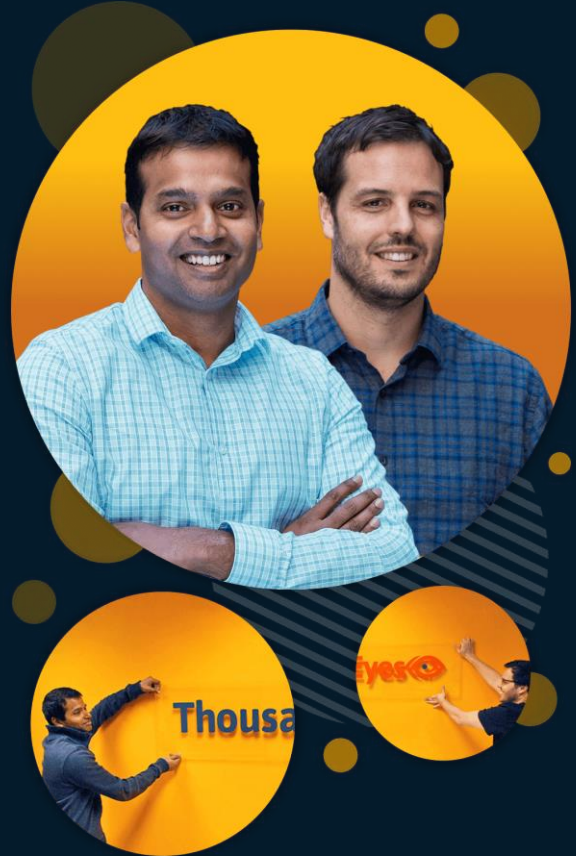
- Enforceable in EU as of 17th January 2025
- Applies to more than 22,000 financial services organizations operational in the EU
- Essentially that financial institutions maintain a resilient ICT framework

What is ThousandEyes



A bet on the future of business!

- Founded in 2010 by Mohit Lad and Ricardo Oliveira, Ph.D students at the UCLA Internet Research Lab
- Set up shop in a garage in Mountain View
- Acquired by Cisco in 2020
- Google Maps for the Internet



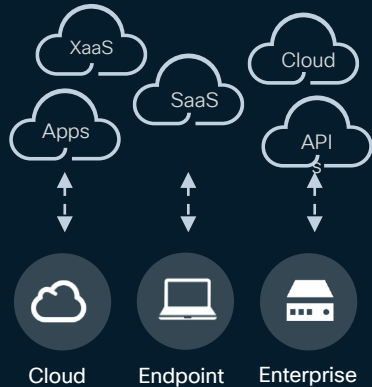
ThousandEyes Internet and Cloud Intelligence

VISIBILITY

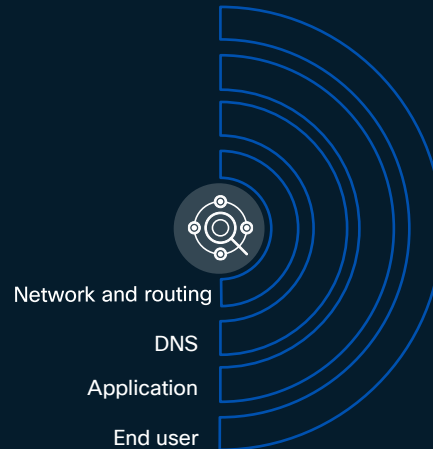
INTELLIGENCE

WORKFLOWS

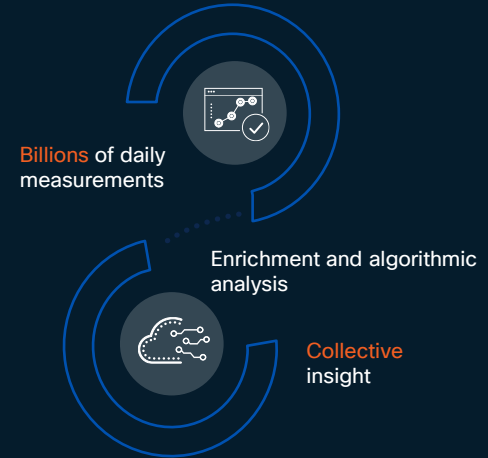
1000s of Global Vantage Points



Unique X-layer Telemetry



Global Collective Insight



ThousandEyes Internet and Cloud Intelligence

VISIBILITY

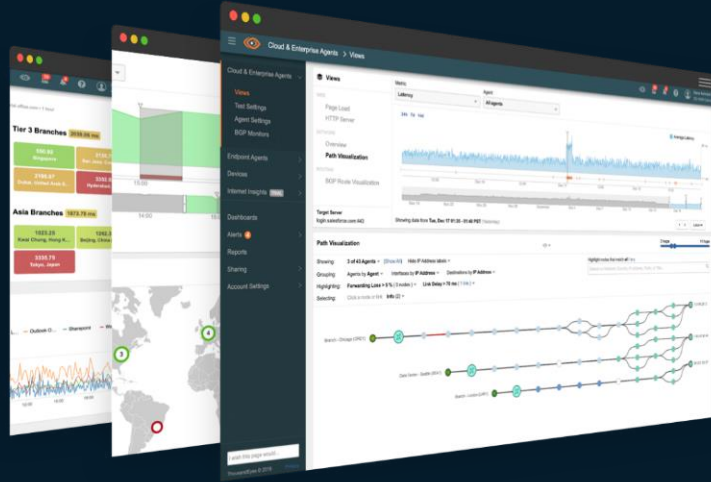


Surface fault domain, root cause, and attribution to responsible party



See digital experience in context with deep correlations across layers

INTELLIGENCE



WORKFLOWS



Gain global insight into Internet and SaaS app outages



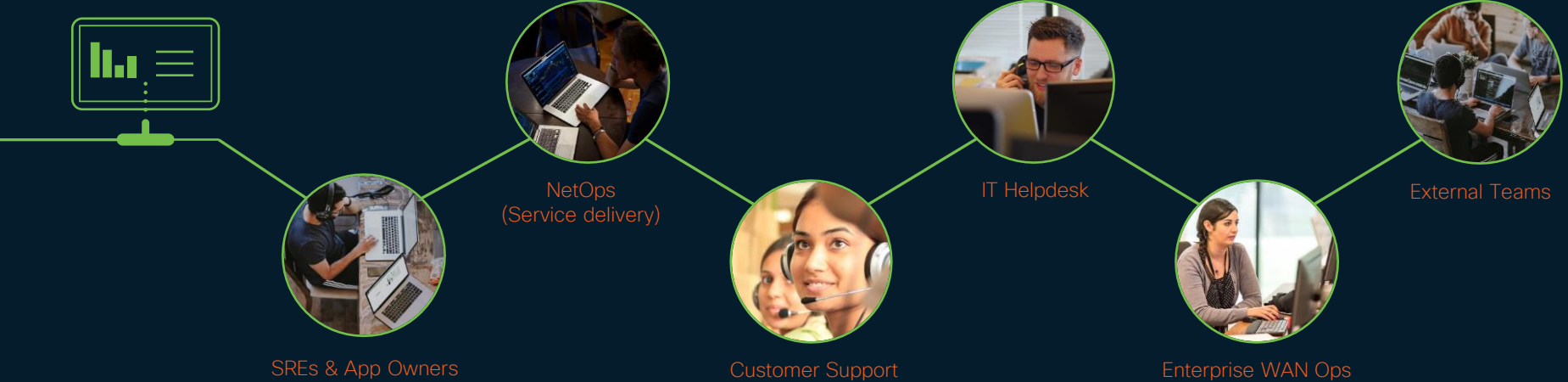
Drill down to quality of experience of individual users and groups in your workforce

ThousandEyes Internet and Cloud Intelligence

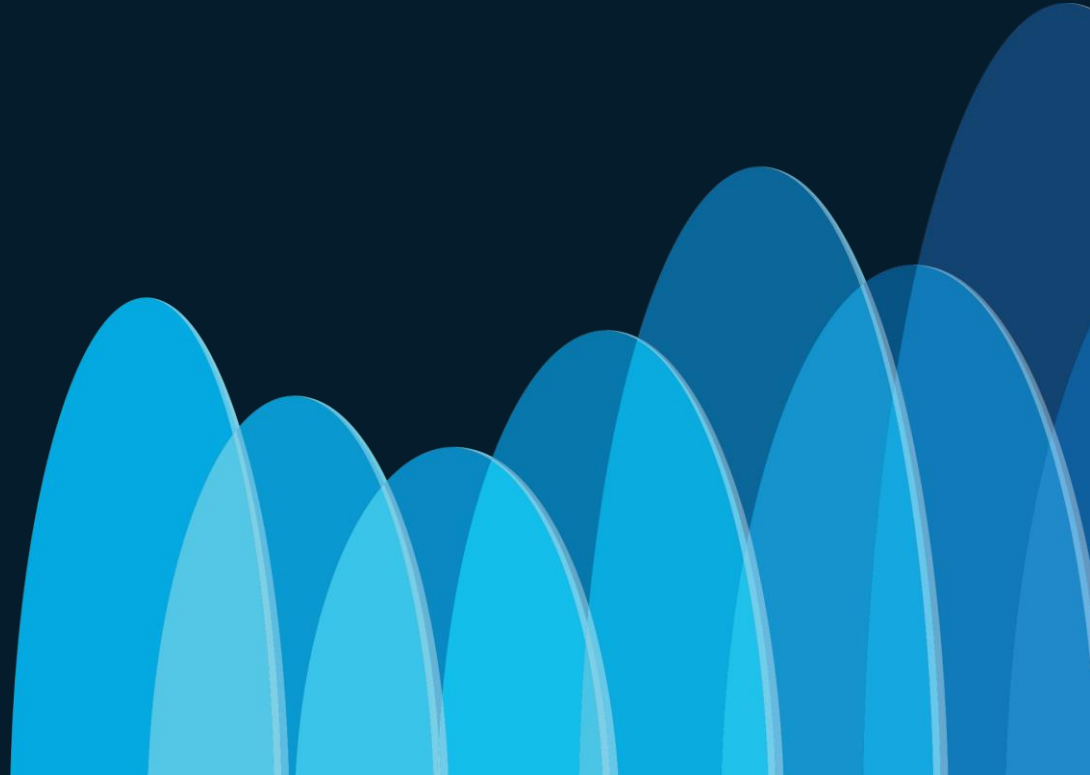
VISIBILITY

INTELLIGENCE

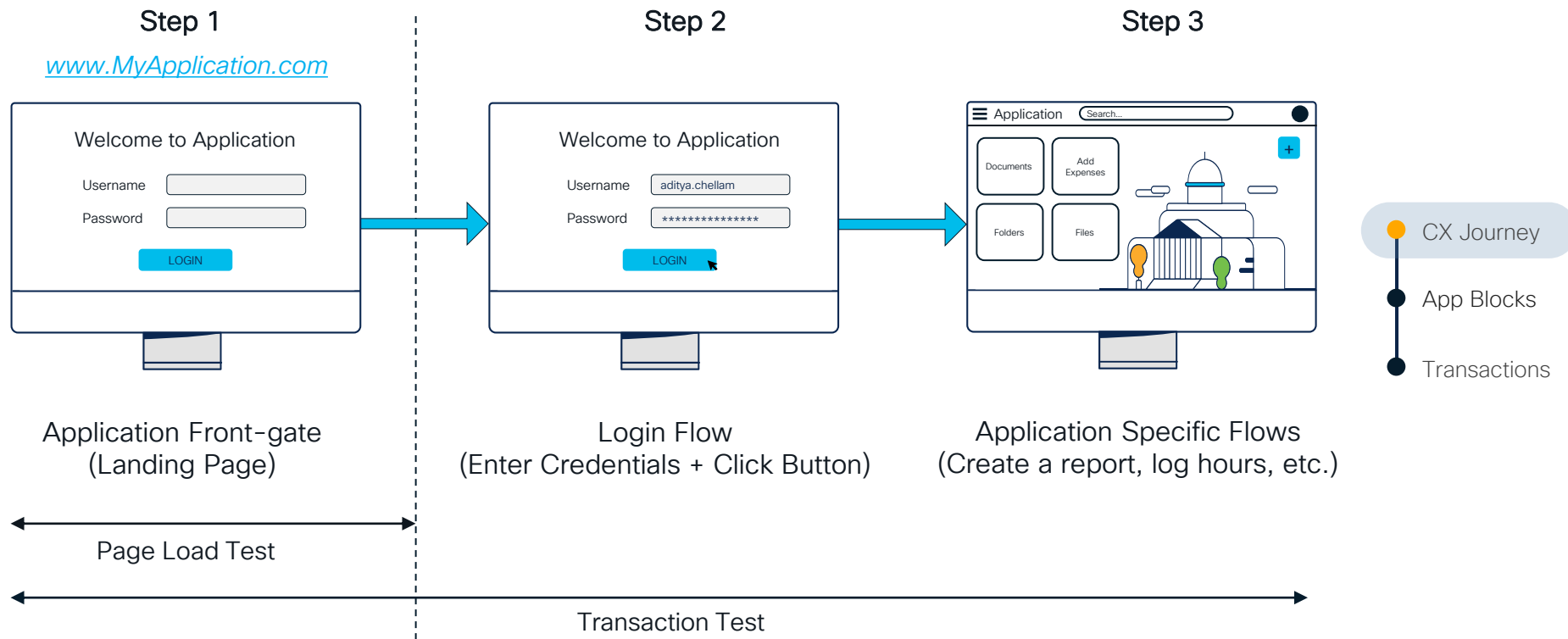
WORKFLOWS



Elevating customer digital experience with Assurance



Understanding the Customer Journey



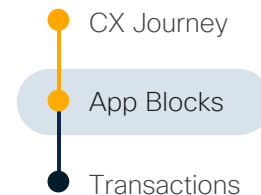
Building Blocks of an Application

Welcome to Application

Username

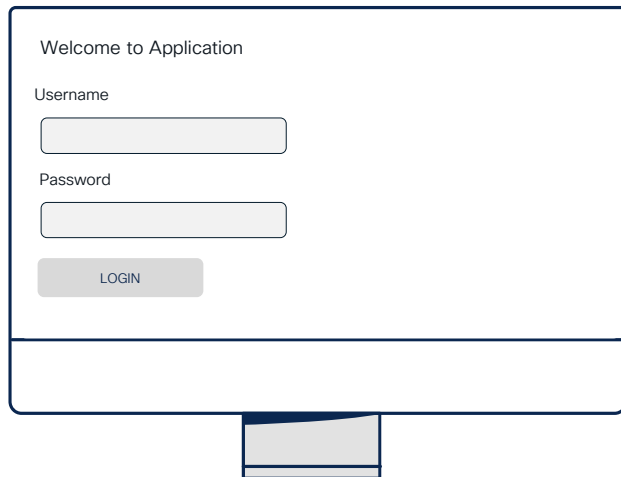
Password

Component	Purpose
HTML	Structures the content of the app
CSS	Styles the app's appearance
JavaScript	Adds interactivity & dynamic functionality
Ajax/XHR	Asynchronous data fetches
Audio	Adds sound effects
Image/Video	Displays visual content



Building Blocks of an Application

HTML - Raw content and structure



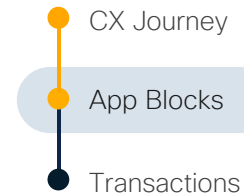
Welcome to Application

Username

Password

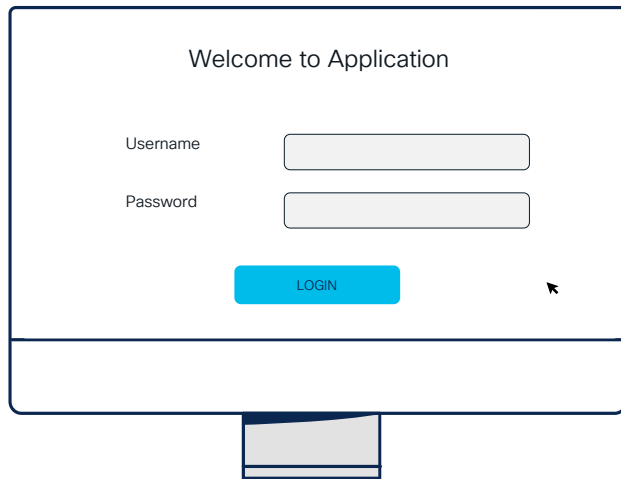
LOGIN

Component	Purpose
HTML	Structures the content of the app
CSS	Styles the app's appearance
JavaScript	Adds interactivity & dynamic functionality
Ajax/XHR	Asynchronous data fetches
Audio	Adds sound effects
Image/Video	Displays visual content



Building Blocks of an Application

CSS – Colors and Styling

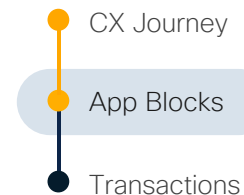


Welcome to Application

Username

Password

Component	Purpose
HTML	Structures the content of the app
CSS	Styles the app's appearance
JavaScript	Adds interactivity & dynamic functionality
Ajax/XHR	Asynchronous data fetches
Audio	Adds sound effects
Image/Video	Displays visual content



Building Blocks of an Application

JavaScript – Intractability and Functionality

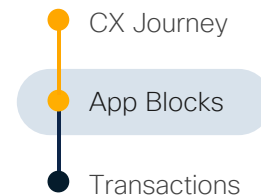
Welcome to Application

Username

Password

LOGIN

Component	Purpose
HTML	Structures the content of the app
CSS	Styles the app's appearance
JavaScript	Adds interactivity & dynamic functionality
Ajax/XHR	Asynchronous data fetches
Audio	Adds sound effects
Image/Video	Displays visual content



Client-Server Communication Model

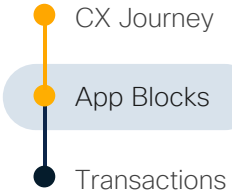
Welcome to Application

Username

Password

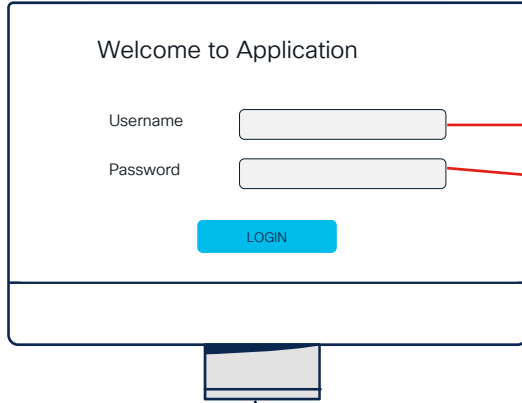
LOGIN

Component	Purpose
HTML	Structures the content of the app
CSS	Styles the app's appearance
JavaScript	Adds interactivity & dynamic functionality
Ajax/XHR	Asynchronous data fetches
Audio	Adds sound effects
Image/Video	Displays visual content



Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	138 ms	320 ms
304	GET	adityachellam.github.io	/	document	html	cached	800 B			
200	GET	adityachellam.github.io	styles.css	stylesheet	css	cached	607 B		0 ms	
200	GET	adityachellam.github.io	script.js	script	js	cached	20 B		0 ms	
404	GET	adityachellam.github.io	favicon.ico	FaviconLoader.sys...	html	cached	9.38 kB		0 ms	

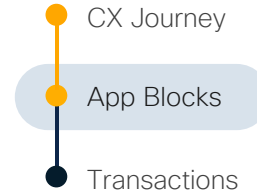
Document Object Model (DOM)



The screenshot shows the Chrome DevTools DOM Inspector. The HTML structure is as follows:

```
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  <body> flex
    <div id="login-container">
      <h1 id="welcome-text">Welcome to Application</h1>
      <form id="login-form"> flex
        <label id="username-label" for="username">Username</label>
        <input id="username" type="text" name="username">
        <br>
        <label id="password-label" for="password">Password</label>
        <input id="password" type="password" name="password">
        <br>
        <button id="login-button" type="submit">LOGIN</button>
      </form>
    </div>
  </body>
</html>
```

The breadcrumb at the bottom of the DOM tree is: `html > body > div#login-container > form#login-form > input#username`. Red boxes highlight the `<input id="username" type="text" name="username">` and `<input id="password" type="password" name="password">` elements, which are pointed to by red arrows from the UI diagram.



The screenshot shows the Chrome DevTools Network tab. The table of network requests is as follows:

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	138 ms	320 ms
304	GET	adityachellam.github.io	/	document	html	cached	800 B			
200	GET	adityachellam.github.io	styles.css	stylesheet	css	cached	607 B		0 ms	
200	GET	adityachellam.github.io	script.js	script	js	cached	20 B		0 ms	
404	GET	adityachellam.github.io	favicon.ico	FaviconLoader.sys...	html	cached	9.38 kB		0 ms	

A red box highlights the first row (Status 304, Method GET, Domain adityachellam.github.io, File /, Initiator document, Type html). A red arrow points from this row to the DOM tree in the image above.

Document Object Model (DOM)

Welcome to Application

Username

Password

LOGIN

Inspector

Console

Debugger

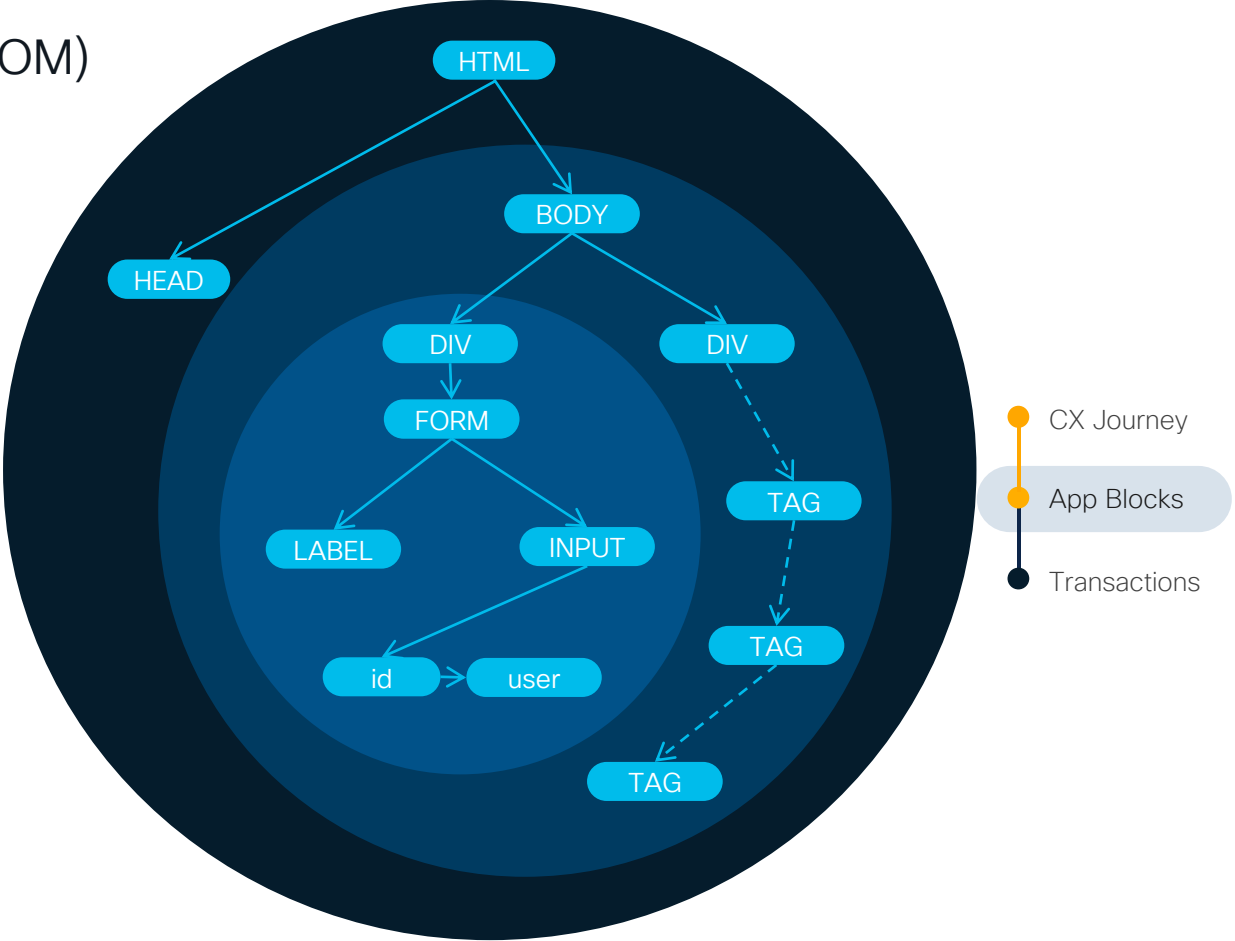
Network

Style Editor

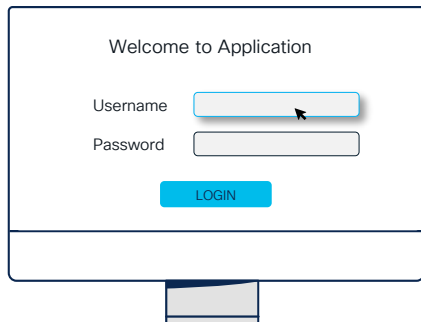
Q Search HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body> (flex)
    <div id="login-container">
      <h1 id="welcome-text">Welcome to Application</h1>
      <div id="login-form"> (flex)
        <label id="username-label" for="username">Username</label>
        <input id="username" type="text" name="username">
        <br>
        <label id="password-label" for="password">Password</label>
        <input id="password" type="password" name="password">
        <br>
        <button id="login-button" type="submit">LOGIN</button>
      </div>
    </div>
  </body>
</html>
```

html > body > div#login-container > form#login-form > input#username



Document Object Model (DOM)



XML Path Language (Xpath)

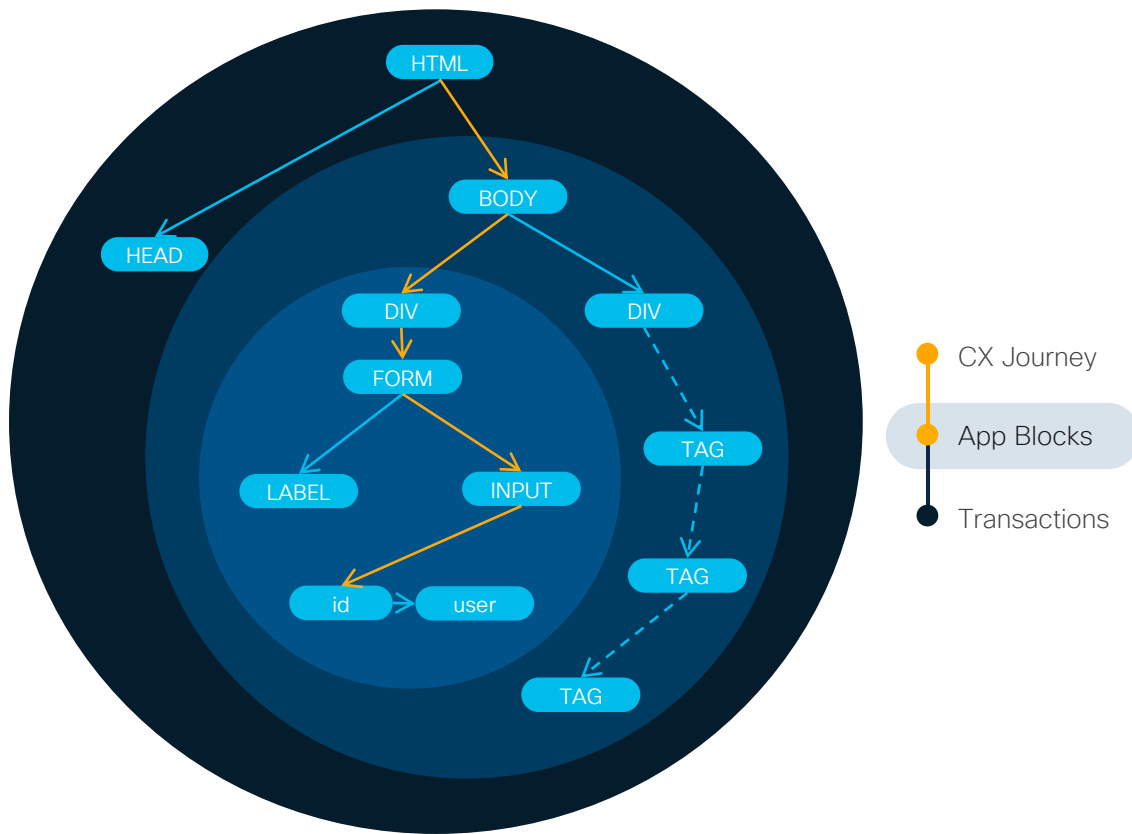
QUERY

Find the INPUT element in the page

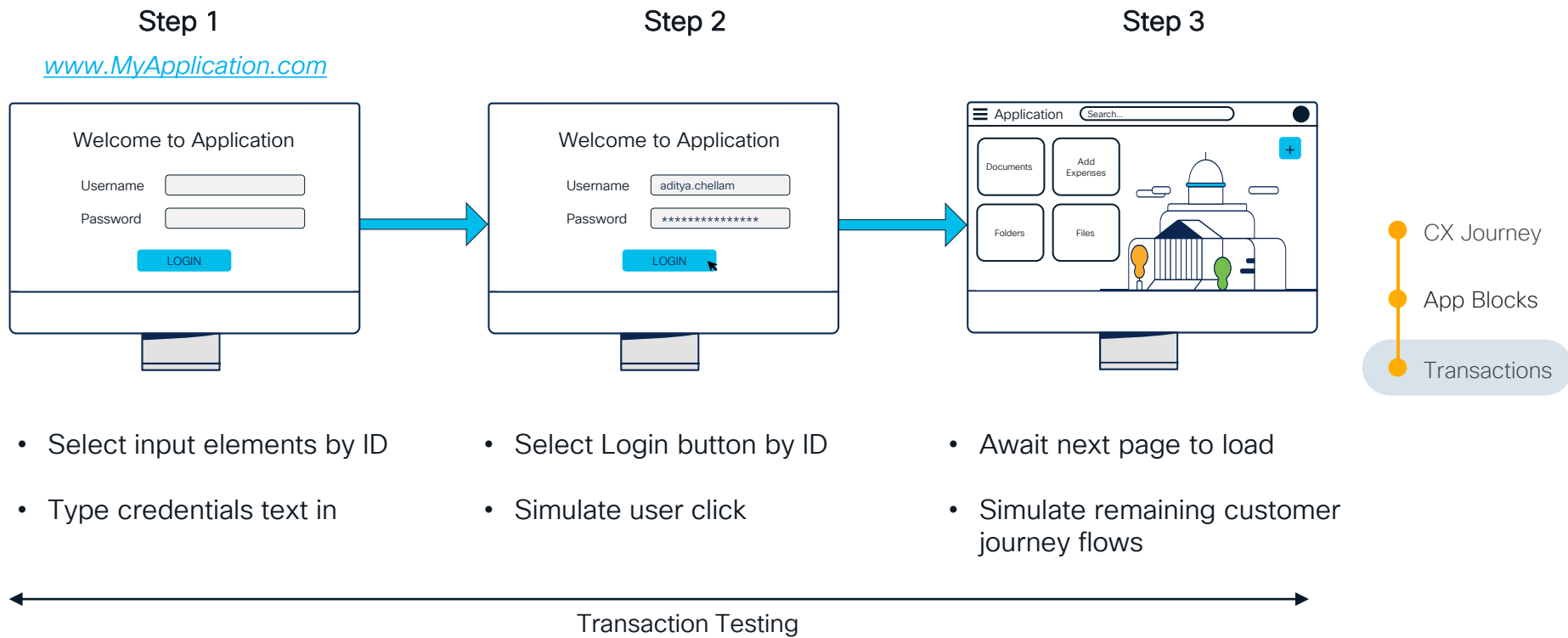
RESULT

/html/body/div[1]/form/input[@id="user"]

OR //*[@id="user"]

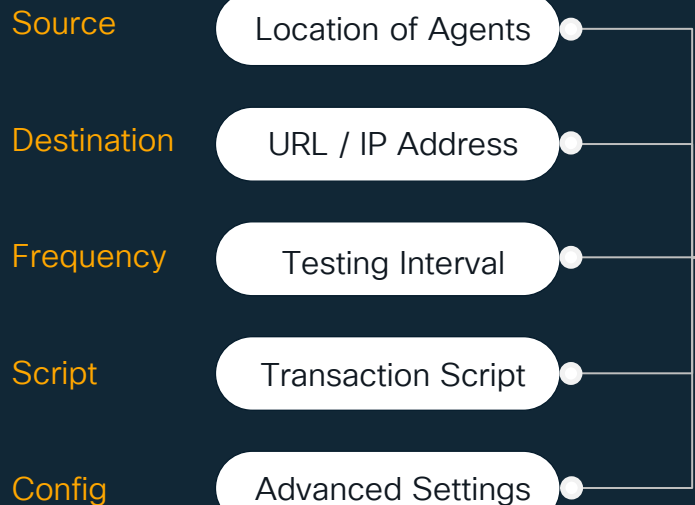


Transaction Test Pseudo-code

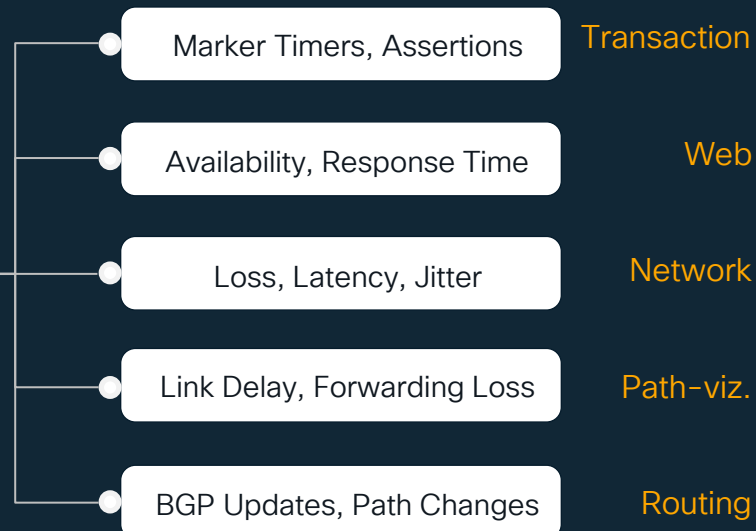


ThousandEyes Setup and Configuration

INPUT



OUTPUT



ThousandEyes Setup and Configuration

INPUT

Basic Configuration **Advanced Settings**

Test Name

SaaS Workflow - Outlook User Experience

Test Description

Optional

Target

https://outlook.office.com/mail/inbox

Schedule

Default Round-Robin

Interval

2 minutes

Agents

3 of 1006 selected

Alerts

☒ Enable

4 of 5 alert rules selected

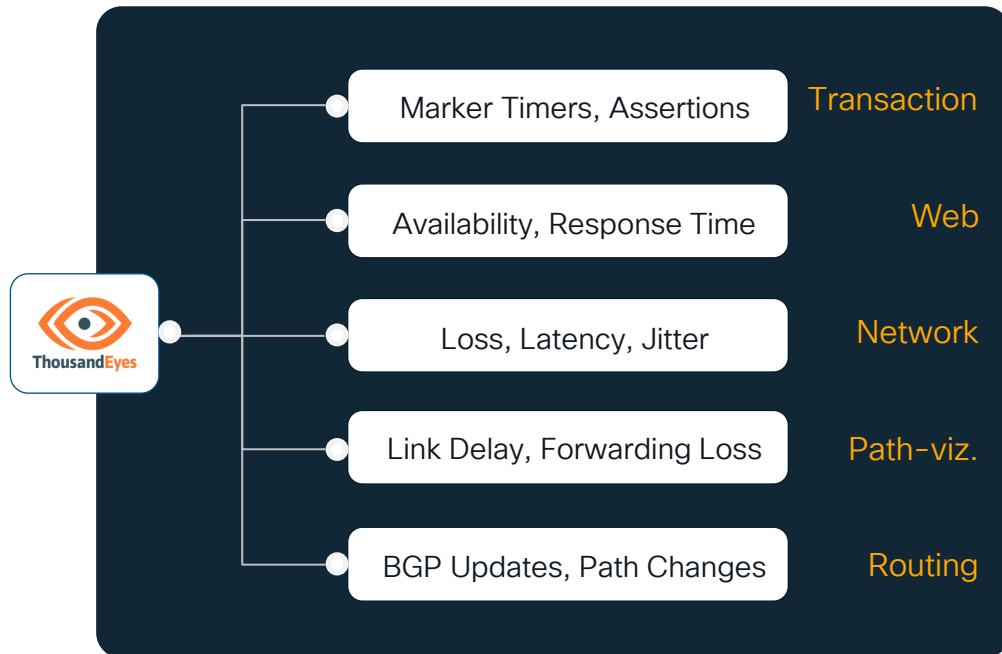
Edit Alert Rules

Transaction Script

Create New

```
1 import { By, Key, until } from 'selenium-webdriver';
2 import { driver, credentials, markers, test } from 'thousandeyes';
3
4 runScript();
5
6 let email = credentials.get('Microsoft_email'); // '<<email>>';
7 let password = credentials.get('Microsoft_password'); // '<<credentialsName>>';
8
9 async function runScript() {
10   await configureDriver();
11   let testSettings = test.getSettings();
12
13   markers.start('FrontGate Load');
```

OUTPUT



ThousandEyes Setup and Configuration

INPUT

Basic Configuration **Advanced Settings**

Test Name

SaaS Workflow - Outlook User Experience

Test Description

Optional

Target

https://outlook.office.com/mail/inbox

Destination

Schedule

Default

Round-Robin

Interval

2 minutes

Interval

Agents

3 of 1006 selected

Source

Alerts





☒ Enable

4 of 5 alert rules selected

Edit Alert Rules

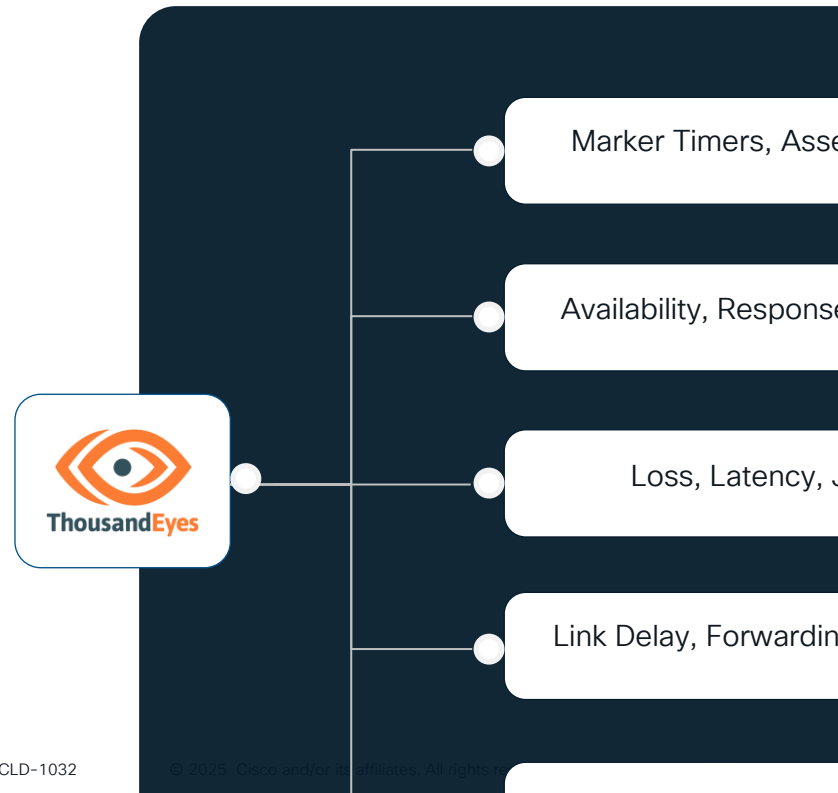
Transaction Script

Create New



```
1 import { By, Key, until } from 'selenium-webdriver';
2 import { driver, credentials, markers, test } from 'thousandeyes';
3
4 runScript();
```

OUTPUT



ThousandEyes Setup and Configuration

INPUT

Basic Configuration **Advanced Settings**

Test Name

SaaS Workflow - Outlook User Experience

Test Description

Optional

Target

https://outlook.office.com/mail/inbox

Schedule

Default Round-Robin

Interval

2 minutes

Agents

3 of 1006 selected

Alerts

☒ Enable

4 of 5 alert rules selected

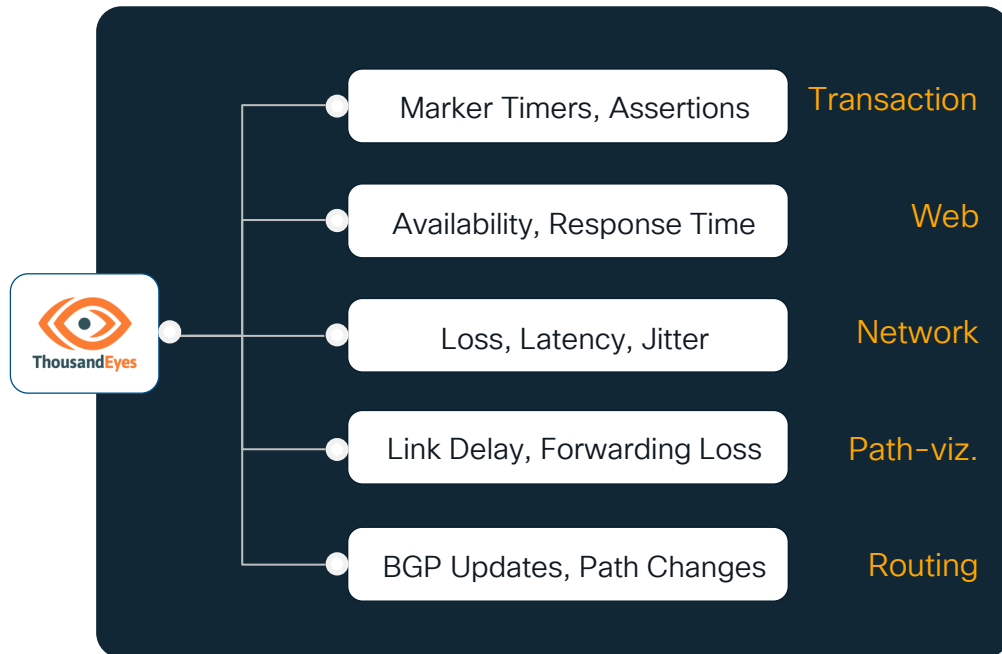
Edit Alert Rules

Transaction Script

Create New

```
1 import { By, Key, until } from 'selenium-webdriver';
2 import { driver, credentials, markers, test } from 'thousandeyes';
3
4 runScript();
5
6 let email = credentials.get('Microsoft_email'); // '<<email>>';
7 let password = credentials.get('Microsoft_password'); // '<<credentialsName>>';
8
9 async function runScript() {
10   await configureDriver();
11   let testSettings = test.getSettings();
12
13   markers.start('FrontGate Load');
```

OUTPUT



ThousandEyes Setup and Configuration

INPUT

Basic Configuration**Advanced Settings**

Test Name

SaaS Workflow - Outlook User Experience

Test Description

Optional

Target

https://outlook.office.com/mail/inbox

Schedule

DefaultRound-Robin

Interval

2 minutes

Agents

3 of 1006 selected

Alerts

☒ Enable

4 of 5 alert rules selected

Edit Alert Rules

Transaction Script

Create New

```
1 import { By, Key, until } from 'selenium-webdriver';
2 import { driver, credentials, markers, test } from 'thousandeyes';
3
4 runScript();
5
6 let email = credentials.get('Microsoft_email'); // '<email>';
7 let password = credentials.get('Microsoft_password'); // '<credentialsName>';
8
9 async function runScript() {
10   await configureDriver();
11   let testSettings = test.getSettings();
12
13   markers.start('FrontGate Load');
```



OUTPUT

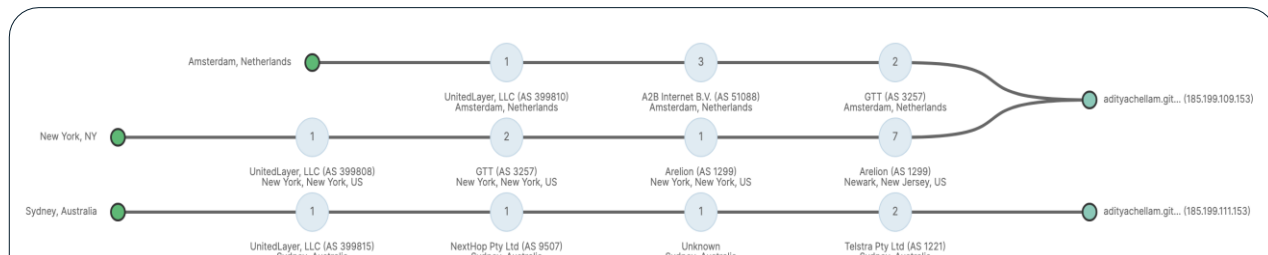


ThousandEyes Setup and Configuration

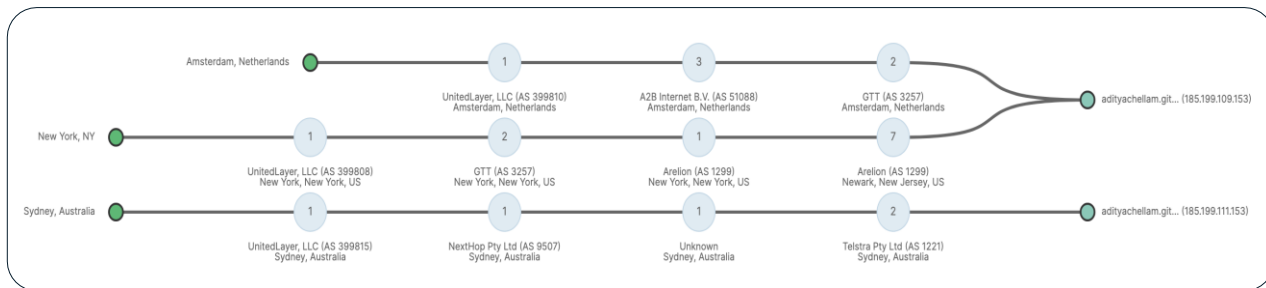
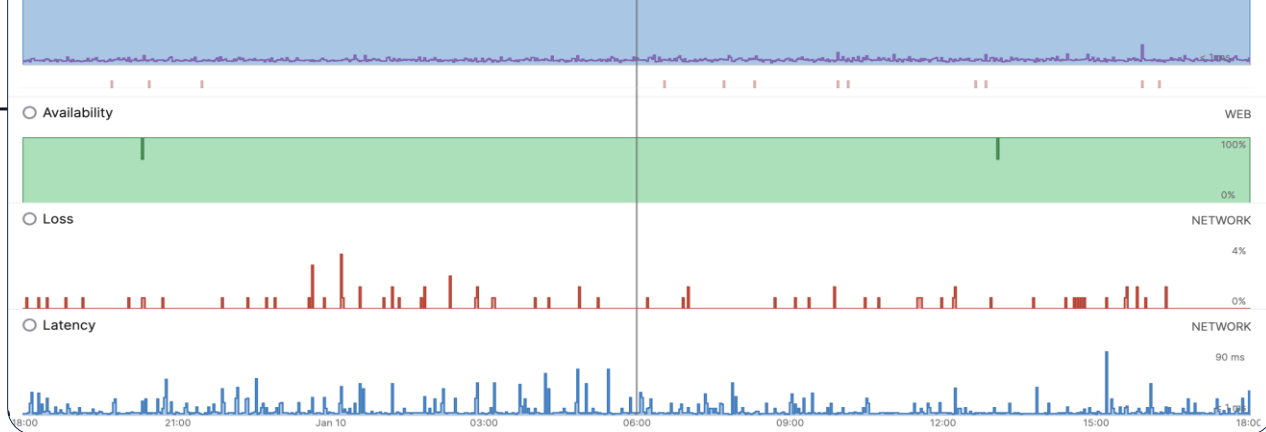
OUTPUT



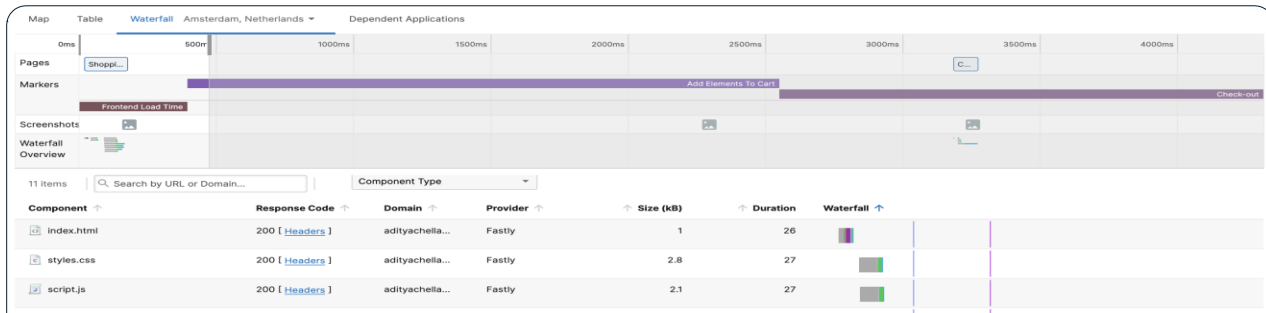
Cross Layer
Correlation

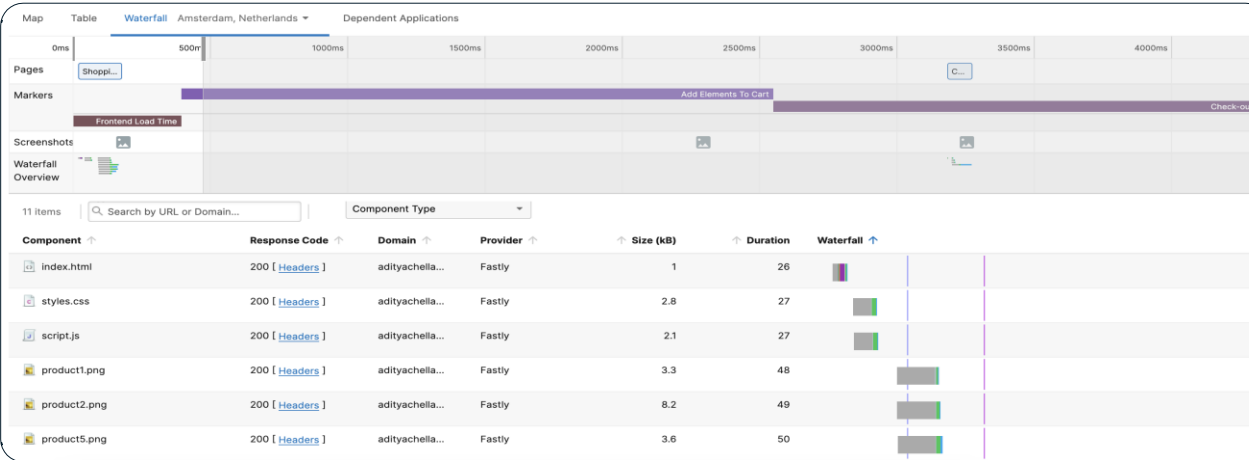
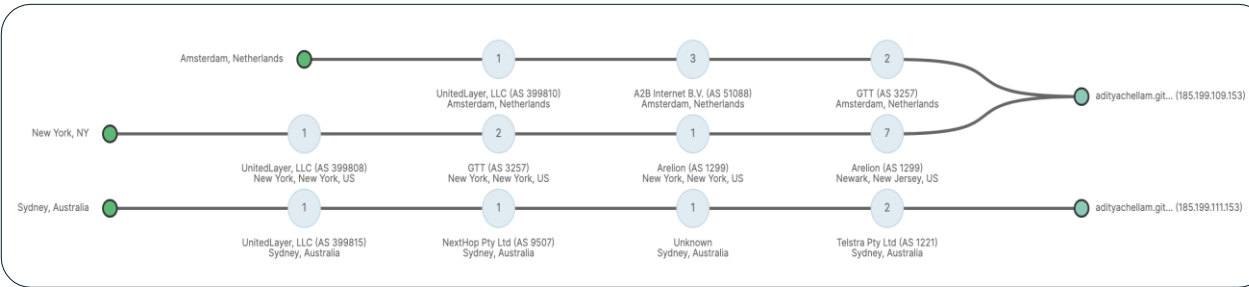


T



Hop-by-hop
path
visualization





Application Layer Insights

ThousandEyes Setup and Configuration

INPUT

Basic Configuration**Advanced Settings**

Test Name

SaaS Workflow - Outlook User Experience

Test Description

Optional

Target

https://outlook.office.com/mail/inbox

Schedule

DefaultRound-Robin

Interval

2 minutes

Agents

3 of 1006 selected

Alerts

☒ Enable

4 of 5 alert rules selected

Edit Alert Rules

Transaction Script

Create New

```
1 import { By, Key, until } from 'selenium-webdriver';
2 import { driver, credentials, markers, test } from 'thousandeyes';
3
4 runScript();
5
6 let email = credentials.get('Microsoft_email'); // '<email>';
7 let password = credentials.get('Microsoft_password'); // '<credentialsName>';
8
9 async function runScript() {
10   await configureDriver();
11   let testSettings = test.getSettings();
12
13   markers.start('FrontGate Load');
```



OUTPUT



ThousandEyes Setup and Configuration

Import required functions to model the transaction

1 `import { driver, markers, credentials } from 'thousandeyes';`

Function Call run by NodeJS backend

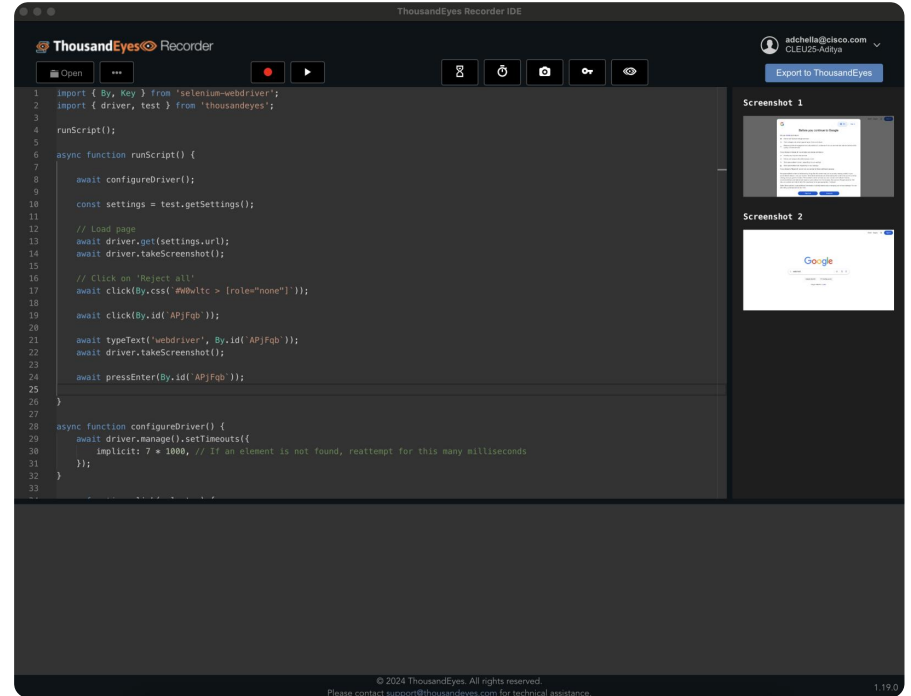
2 `runScript();`

Function Definition specifying step by step interaction with the DOM

3 `async function runScript() {
 const settings = test.getSettings();
 .
 .
}`

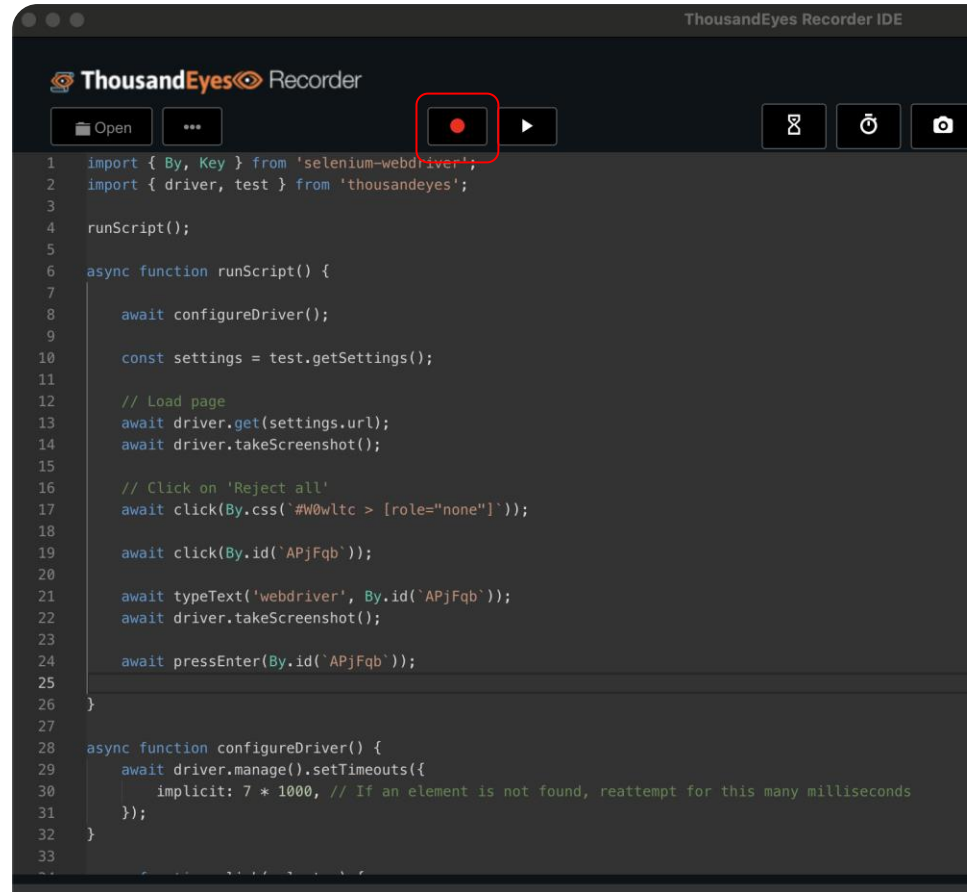
ThousandEyes Recorder IDE

- Automatically convert user interactions to code with the ThousandEyes Recorder IDE



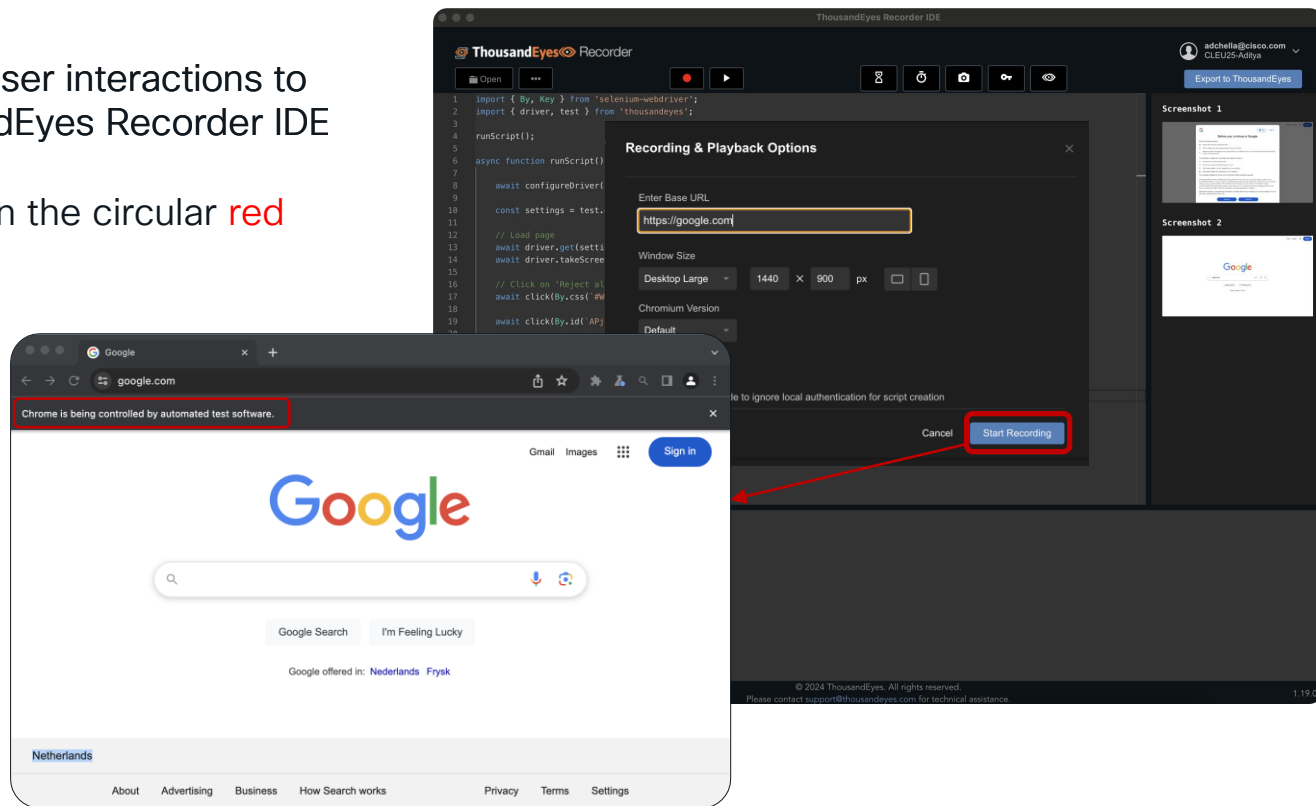
ThousandEyes Recorder IDE

- Automatically convert user interactions to code with the ThousandEyes Recorder IDE
- Hit record by clicking on the circular red record button



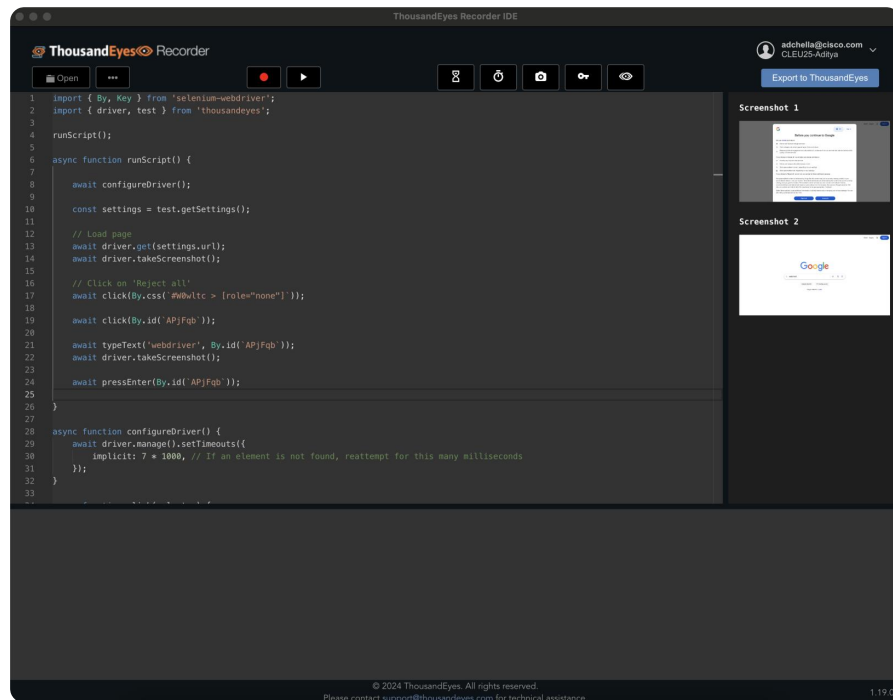
ThousandEyes Recorder IDE

- Automatically convert user interactions to code with the ThousandEyes Recorder IDE
- Hit record by clicking on the circular **red** record button
- Enter the target application URL to interact with the instance of chromium browser opened



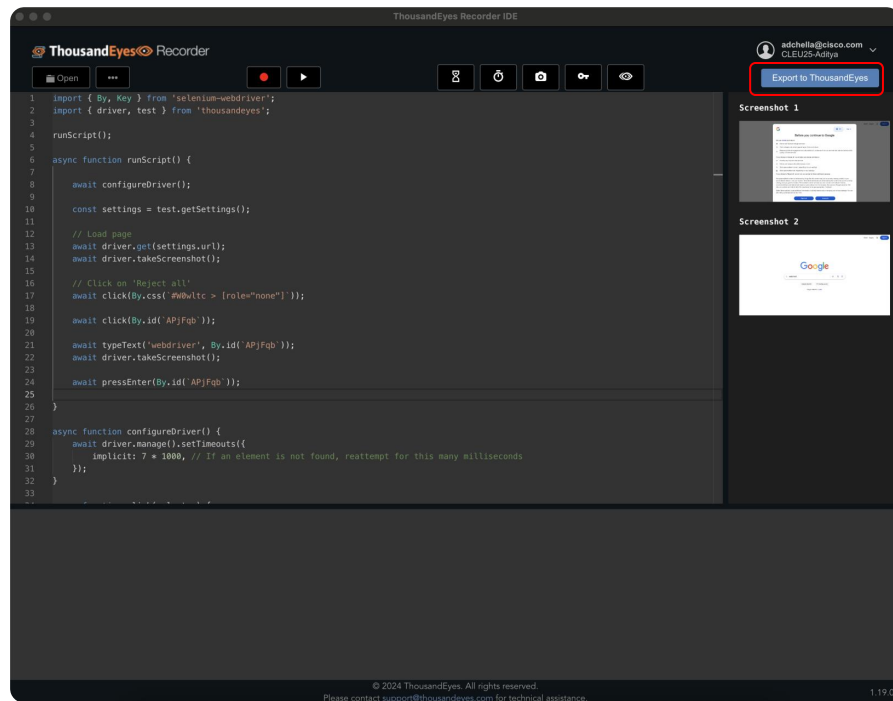
ThousandEyes Recorder IDE

- Automatically convert user interactions to code with the ThousandEyes Recorder IDE
- Hit record by clicking on the circular **red** record button
- Enter the target application URL to interact with the instance of chromium browser opened
- Follow the sequence of steps to be modelled into code

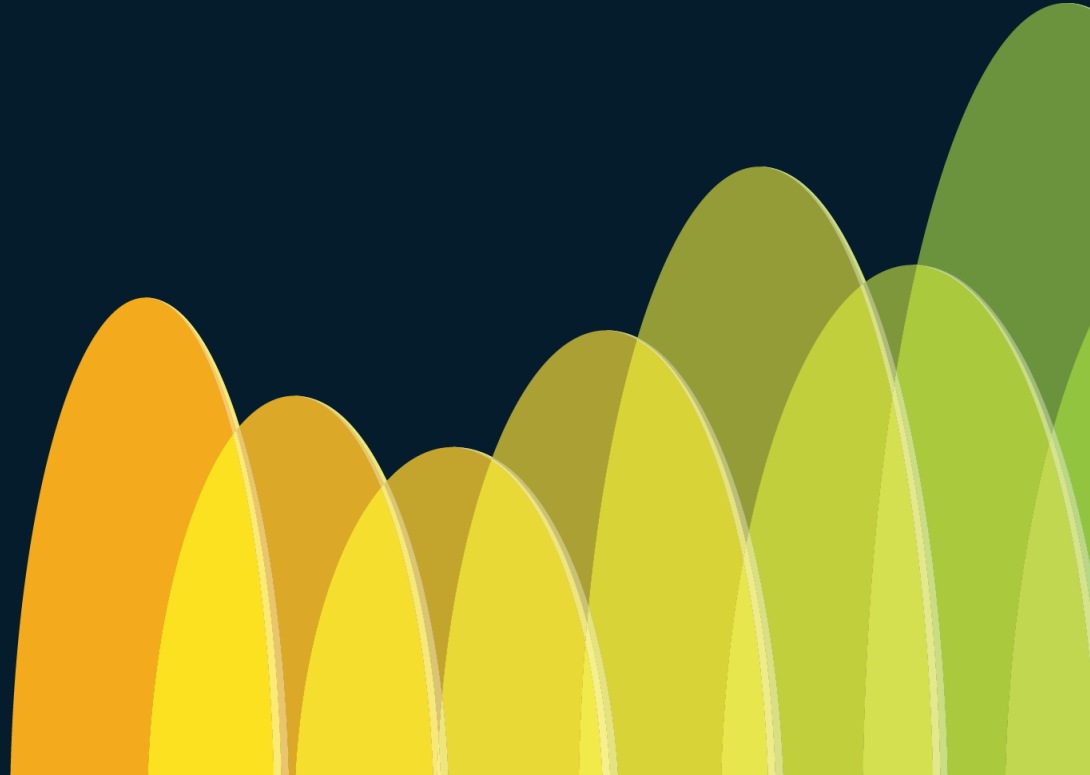


ThousandEyes Recorder IDE

- Automatically convert user interactions to code with the ThousandEyes Recorder IDE
- Hit record by clicking on the circular **red** record button
- Enter the target application URL to interact with the instance of chromium browser opened
- Follow the sequence of steps to be modelled into code
- Export the generated code to the ThousandEyes controller

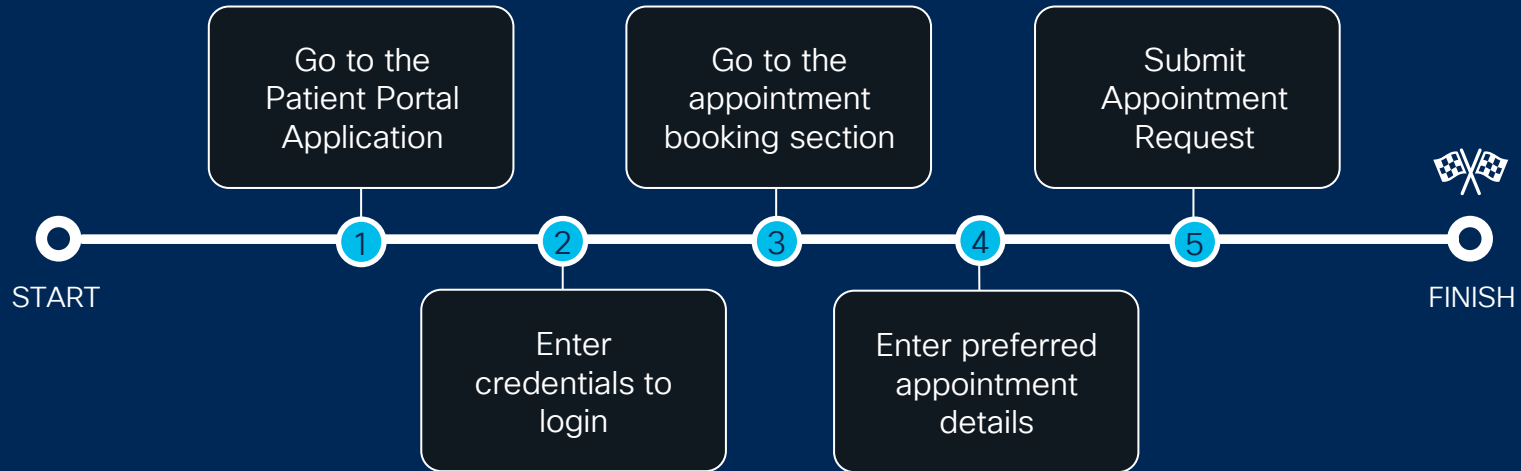


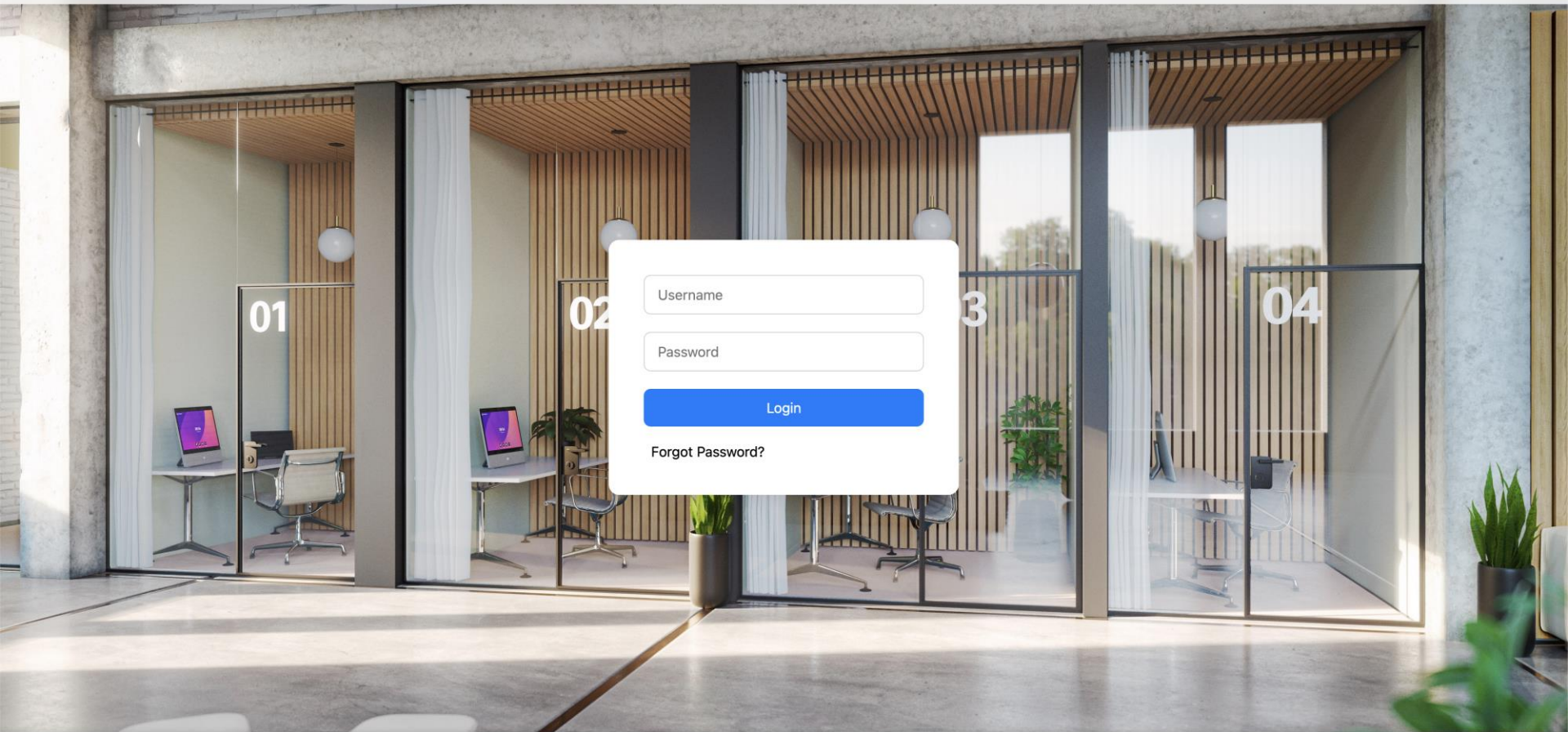
Transaction testing for customer journeys

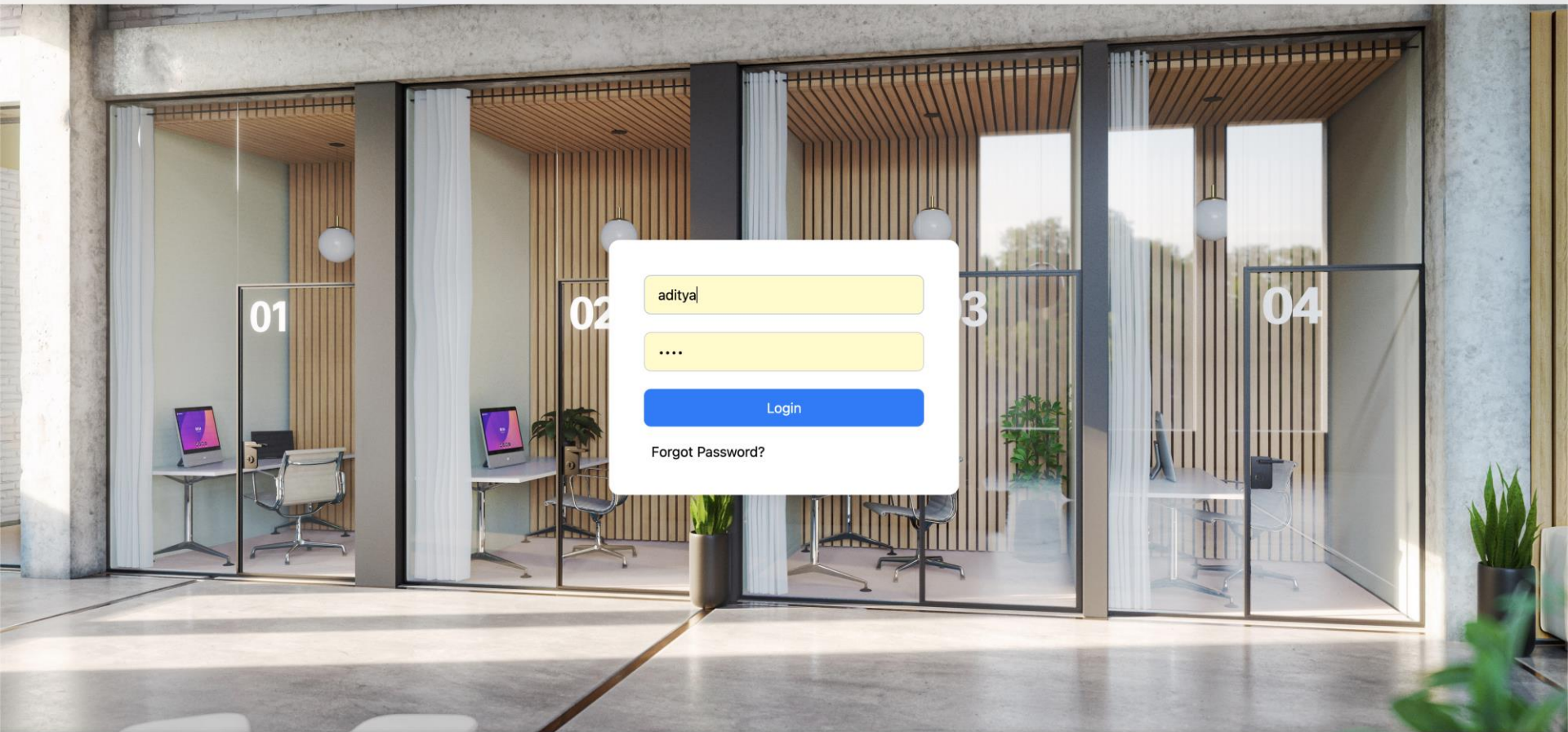


Healthcare : Patient Portal Experience

Patient's User Journey







© 2025 Patient Records System @adityachellam-CLEU25

Patient Overview

- Medical History
- Prescriptions
- Appointments
- Billing

Patient Details

Personal Information

Name: Aditya Chellam
Age: 100
Gender: Male
Blood Group: ABC+

Contact Details

Phone: +1 (234) 123-4567
Email: adchella@cisco.com
Address: 123, cupboard under the staircase, Privet Drive

Recent Medical Records

Annual Checkup - Jan 15, 2025
General health check. All vitals normal.

Blood Test - Dec 5, 2023
Cholesterol levels slightly elevated. Recommended diet plan.

Download Patient Details

Patient Overview

Medical History

Prescriptions

Appointments

Billing

Book an Appointment

Patient Name

Enter preferred name

Select Doctor

Dr. Albus Dumbledore - General Physician

Date

dd / mm / yyyy

Reason for Visit

Regular Checkup

Book Appointment

Available Time Slots

09:00 AM

10:30 AM

11:45 AM

02:00 PM

03:30 PM

04:45 PM

Patient Overview

Medical History

Prescriptions

Appointments

Billing

Book an Appointment

Patient Name

Aditya Chellam

Select Doctor

Dr. Minerva McGonagall - Cardiologist

Date

13 / 02 / 2025

Reason for Visit

Specialist Consultation

Book Appointment

Available Time Slots

09:00 AM

10:30 AM

11:45 AM

02:00 PM

03:30 PM

04:45 PM

Patient Overview

Medical History

Prescriptions

Appointments

Billing

Book an Appointment

Patient Name

Aditya Chellam

Select Doctor

Dr. Minerva McGonagall - Cardiologist

Date

13 / 02 / 2025

Reason for Visit

Specialist Consultation

Book Appointment

Appointment booked for Aditya Chellam on 2025-02-13 with Dr. Minerva McGonagall.

Available Time Slots

09:00 AM

10:30 AM

11:45 AM

02:00 PM

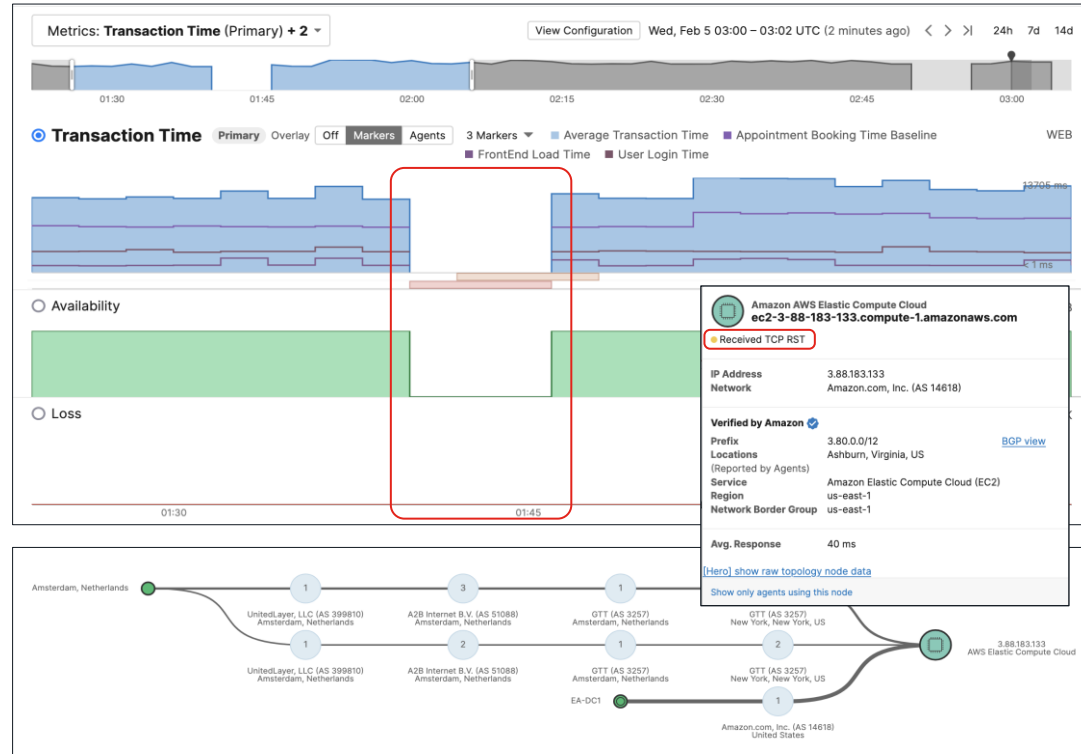
03:30 PM

04:45 PM

Application Service Not Running

ANALYSIS

The application HTTP server availability dropped to 0%, yet there was no corresponding spike in packet loss. This suggests that the underlying network connectivity to the server remains intact, reducing the troubleshooting radius to the application layer.



Issue 1

Issue 2






Issue 3

API Call Errors deeper down the journey






ANALYSIS

ThousandEyes transaction tests detected a critical issue in the appointment booking flow—an unintended change from a POST to a GET request for the **Submit Appointment** button, causing a 405 Method Not Allowed error. The **waterfall view** pinpointed the exact failure, showcasing how transaction testing goes beyond uptime monitoring to catch real user-impacting misconfigurations early.

Working Scenario

 appointments/	200 [Headers]	3.88.183.133	5.4	167
 appointments.css	200 [Headers]	3.88.183.133	3.3	164
 jquery-3.6.0.min.js	200 [Headers]	code.jquery.c... Fastly	89.5	31
 bg_img_compressed1.png	200 [Headers]	3.88.183.133	4892.9	0
 book-appointment/	201 [Headers]	3.88.183.133	0.1	168

Non-working Scenario

 appointments/	200 [Headers]	3.88.183.133	5.4	167
 appointments.css	200 [Headers]	3.88.183.133	3.3	163
 jquery-3.6.0.min.js	200 [Headers]	code.jquery.c... Fastly	89.5	35
 bg_img_full1.png	200 [Headers]	3.88.183.133	13738.4	1070
 book-appointment/	405 [Headers]	3.88.183.133	0.1	166

- Issue 1
- Issue 2
- Issue 3

Improving UX through Resource Optimization

ANALYSIS

ThousandEyes transaction testing identified **uncompressed background images** as a performance bottleneck, increasing load times. The **waterfall view** pinpointed these oversized assets, and compression significantly improved performance, enhancing user experience. This showcases ThousandEyes' ability to uncover front-end inefficiencies beyond basic availability monitoring.

Component ↑	Response Code ↑	Domain ↑	Provider ↑	Size (kB) ↑	Duration ↑	Waterfall ↑
appointments/	200 [Headers]	3.88.183.133		5.4	167	
appointments.css	200 [Headers]	3.88.183.133		3.3	166	
jquery-3.6.0.min.js	200 [Headers]	code.jquery.c...	Fastly	89.5	24	
bg_img_full1.png	200 [Headers]	3.88.183.133		13738.4	1116	

Component ↑	Response Code ↑	Domain ↑	Provider ↑	Size (kB) ↑	Duration ↑	Waterfall ↑
appointments/	200 [Headers]	3.88.183.133		5.4	167	
appointments.css	200 [Headers]	3.88.183.133		3.3	180	
jquery-3.6.0.min.js	200 [Headers]	code.jquery.c...	Fastly	89.5	41	
bg_img_compressed1.png	200 [Headers]	3.88.183.133		4892.9	0	

Average Metrics

Transaction Time	13325 ms
FrontEnd Load Time	1004 ms
User Login Time	3700 ms
Appointment Bookin...	8545 ms

Average Metrics

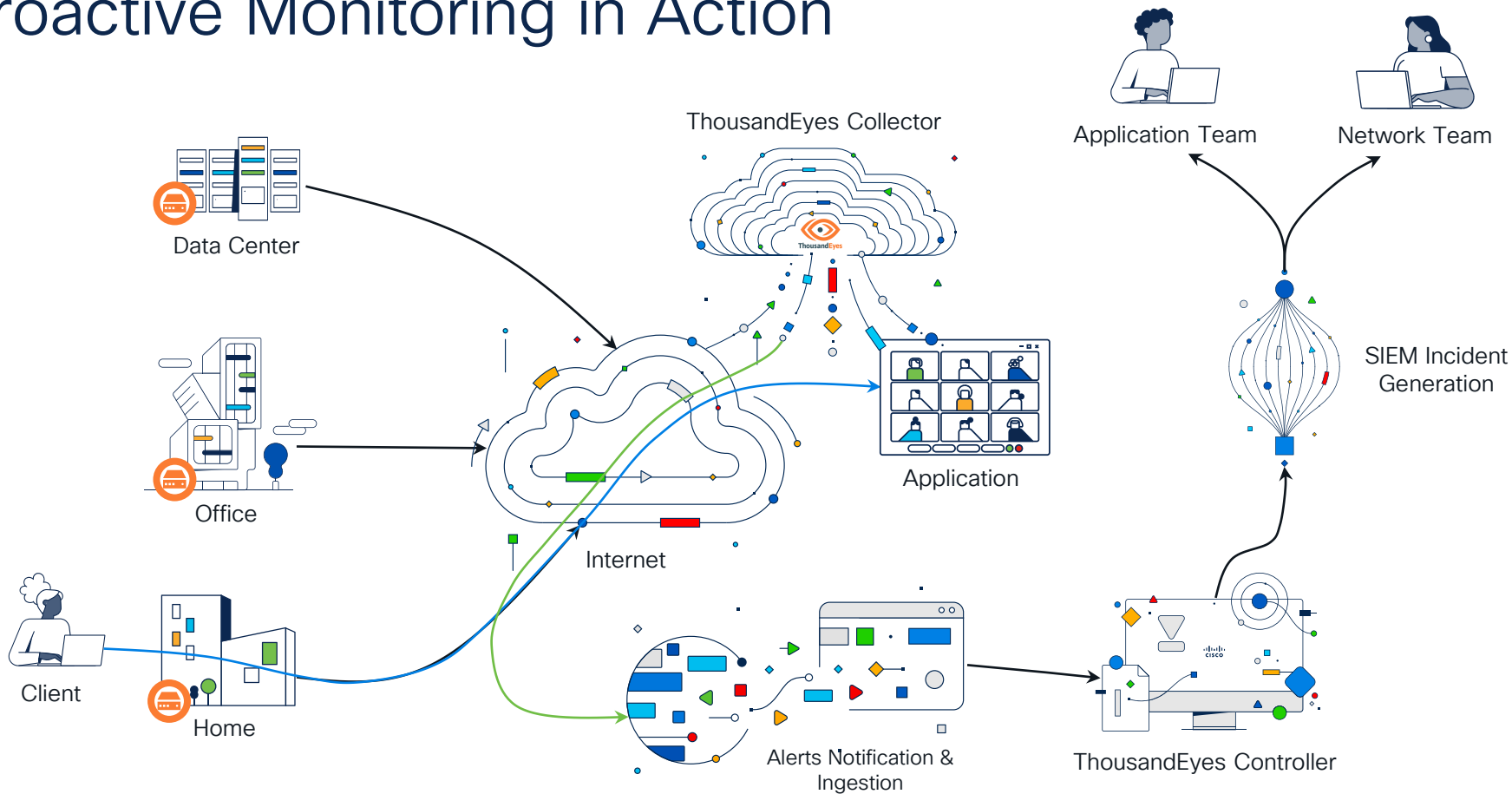
Transaction Time	12005 ms
FrontEnd Load Time	1002 ms
User Login Time	3017 ms
Appointment Bookin...	7920 ms

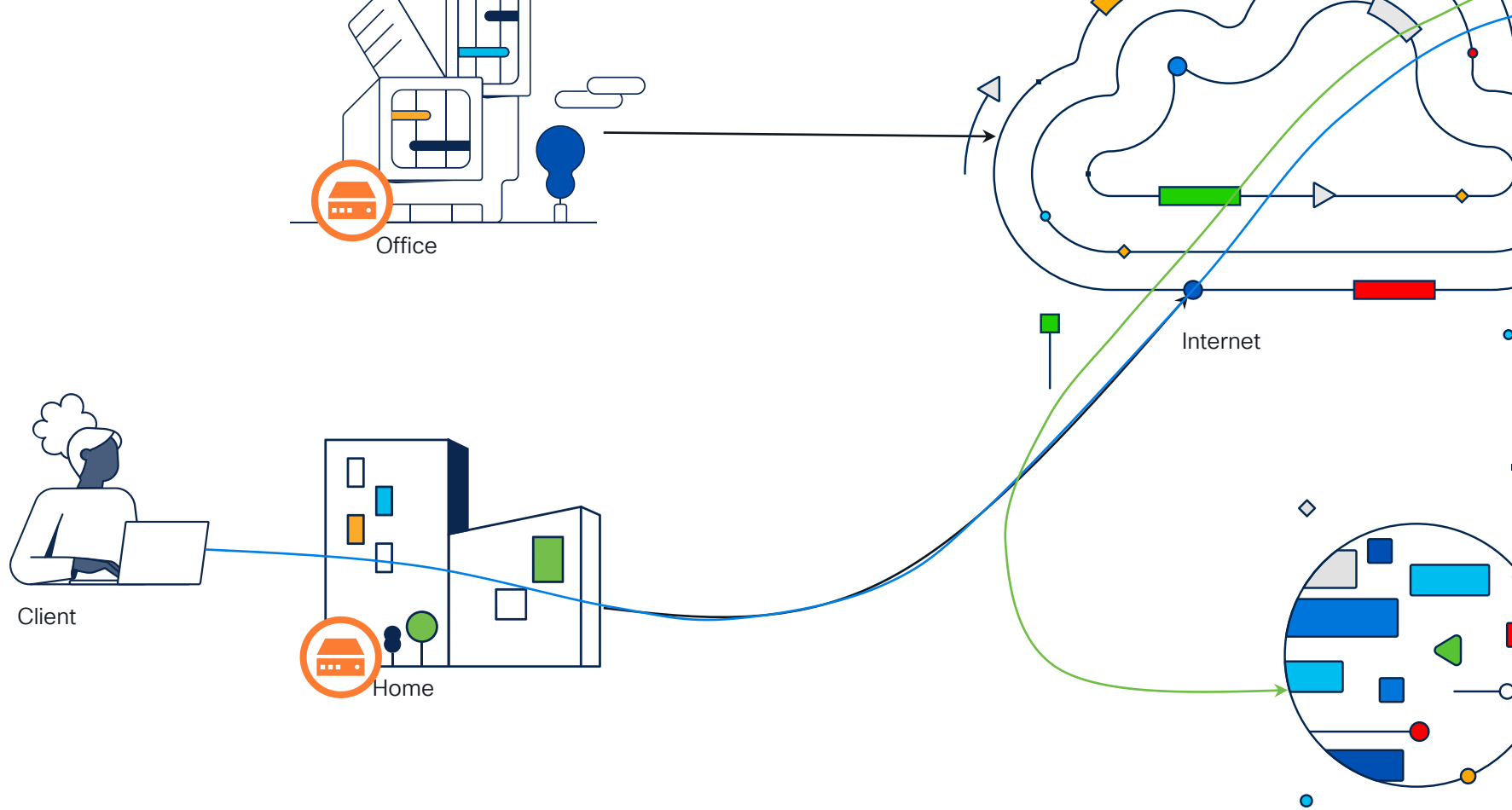
- Issue 1
- Issue 2
- Issue 3

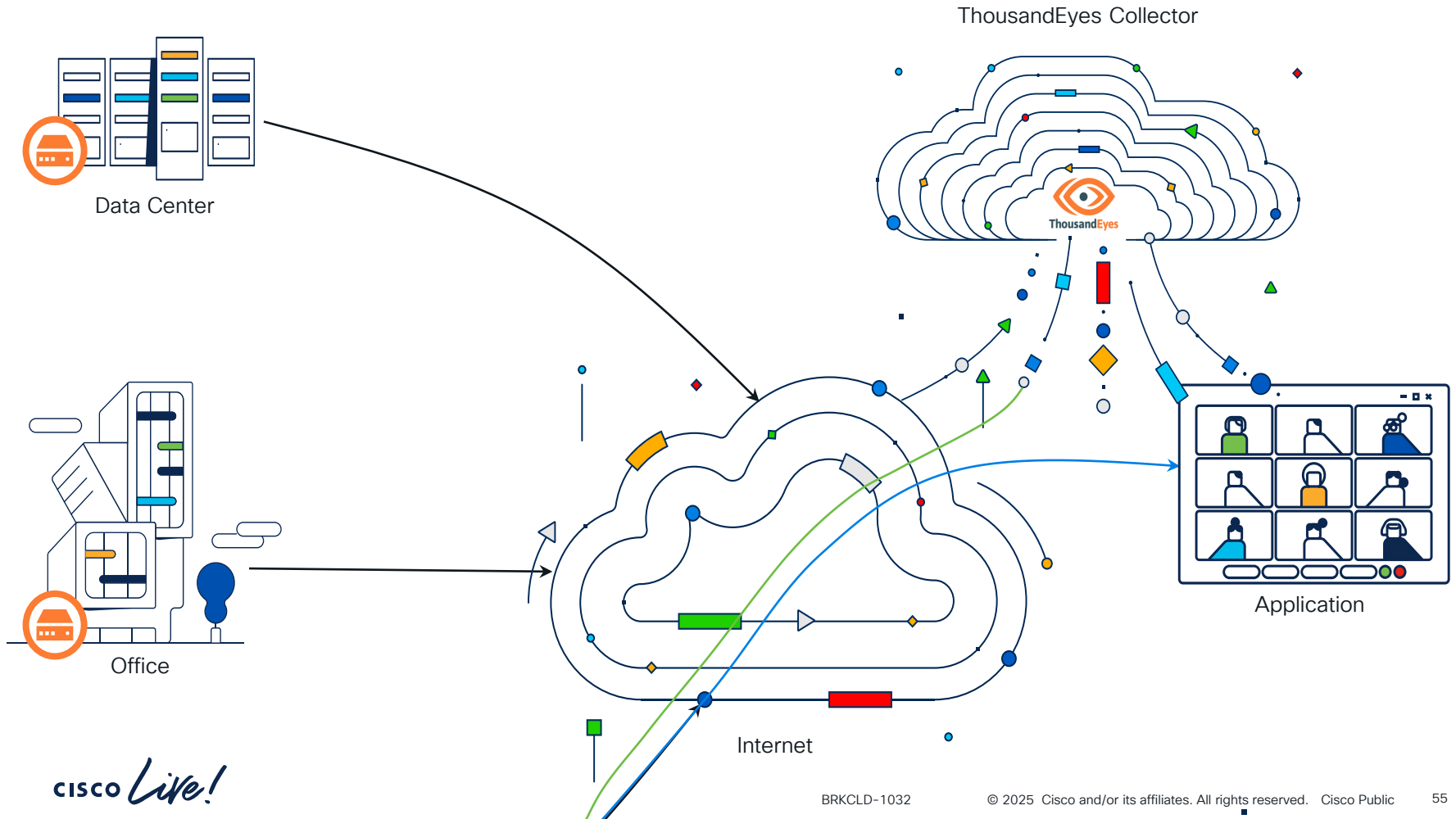
Proactive Monitoring in Action



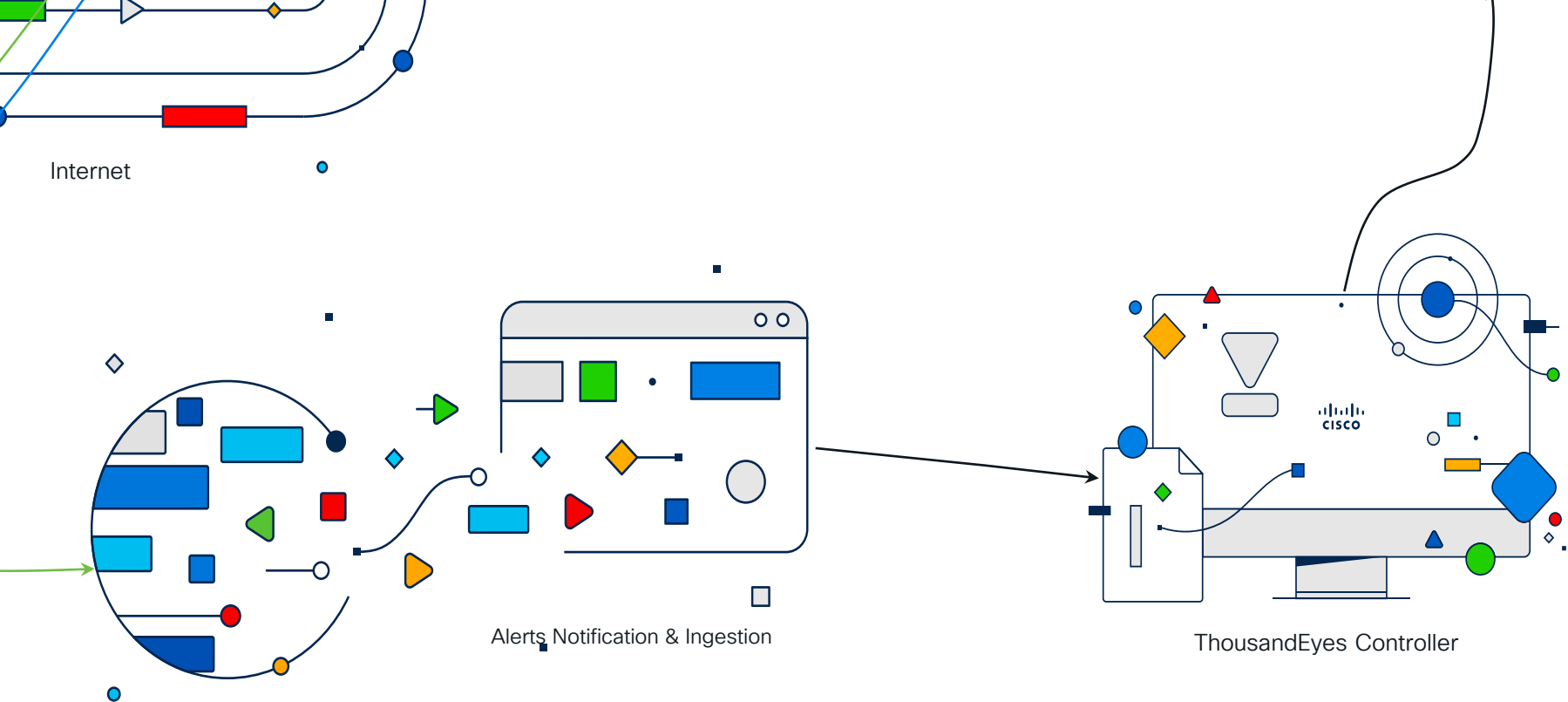
Proactive Monitoring in Action







Internet



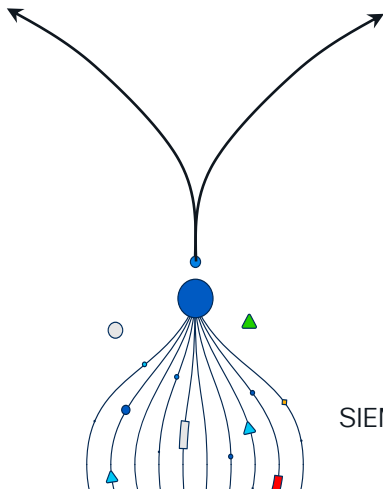
Proactive Monitoring in Action



Application Team



Network Team



SIEM Incident Generation

Drastic **Reduction** in -

- Mean Time To Identification (MTTI)
- Mean Time To Resolution (MTTR)
- Customer Reporting of issues to service desk

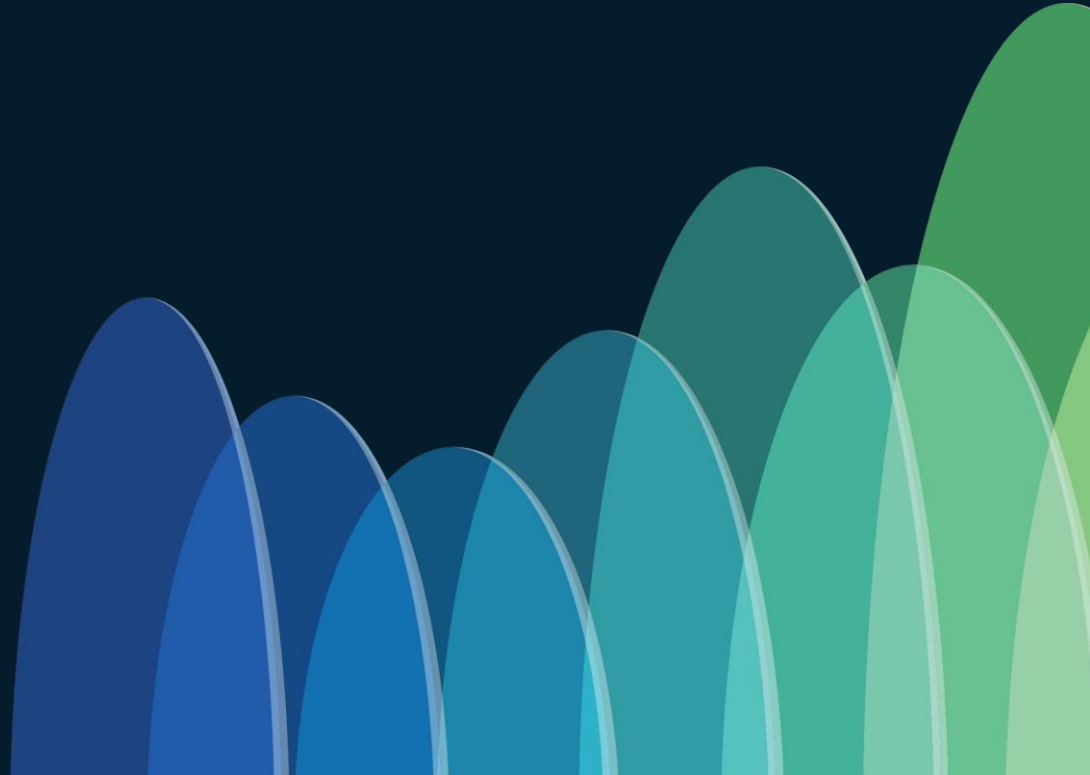
Significant **Increase** in -

- Trust on Service Provider
- Brand Reputation
- Observability into 3rd party performance



CISCO *Live!*

Summary and Key Takeaways



Summary and Key Takeaways

- Digital Complexity Demands Precision: Modern applications rely on interconnected systems, APIs, and third-party services, creating a fragile yet critical digital ecosystem.
- Why Basic Monitoring Fails: Traditional methods miss nuanced user flows and dependencies that impact digital experience.
- ThousandEyes Transaction Testing: Simulates user journeys, providing end-to-end visibility into multi-step workflows, identifying bottlenecks, and enabling proactive issue resolution.
- Real-World Impact: From banking to e-commerce, transaction testing ensures operational resilience, safeguards revenue, and enhances user trust.
- Regulatory Alignment: With frameworks like DORA, the ability to monitor, report, and test digital workflows is no longer optional—it's mandatory.

Continue your education

CISCO *Live!*

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at ciscolive.com/on-demand. Sessions from this event will be available from March 3.

Contact me at: adchella@cisco.com

Fill Out Your Session Surveys



Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)



All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the 'Participant Dashboard'



Content Catalog



Thank you



CISCO *Live!*

GO BEYOND

The background of the slide features a series of overlapping, teardrop-shaped elements in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are arranged in a way that creates a sense of depth and movement, resembling a stylized mountain range or a series of waves. The overall aesthetic is clean and modern.