

8ème édition de la
Soirée du Test
Logiciel
Sophia -
Antipolis



13 octobre 2025
15h à 22h30



Amadeus, Biot



Soirée du
test logiciel

✓ Sophia Antipolis

Moderniser vos tests de performance

l'IA, une réalité à exploiter dès aujourd'hui



Thomas Ripoche
VP - NeoLoad Global Customer Growth



Agenda

Qui est Tricentis ?

Les grands défis de validation de la performance aujourd'hui

NeoLoad – exploitez l'Intelligence Artificielle sans attendre

Questions et réponses

We are **Tricentis**.

A global leader in quality engineering

RESULTS

10x **90%** **50%**

Faster releases

Risk coverage

Cost reduction

PARTNERS

accenture Capgemini Cognizant

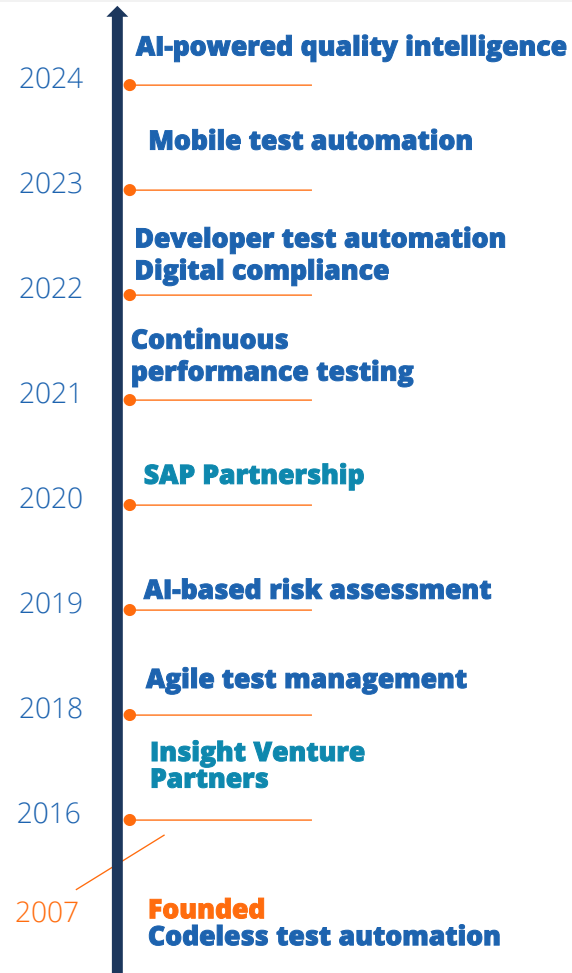
✓ttc Infosys IBM wipro

TECH ALLIANCES

ATLASSIAN aws Microsoft

SAP salesforce servicenow

Driven by growth and innovation



CUSTOMERS

3k+

Customers



ANALYST RECOGNITION

Gartner®

FORRESTER®

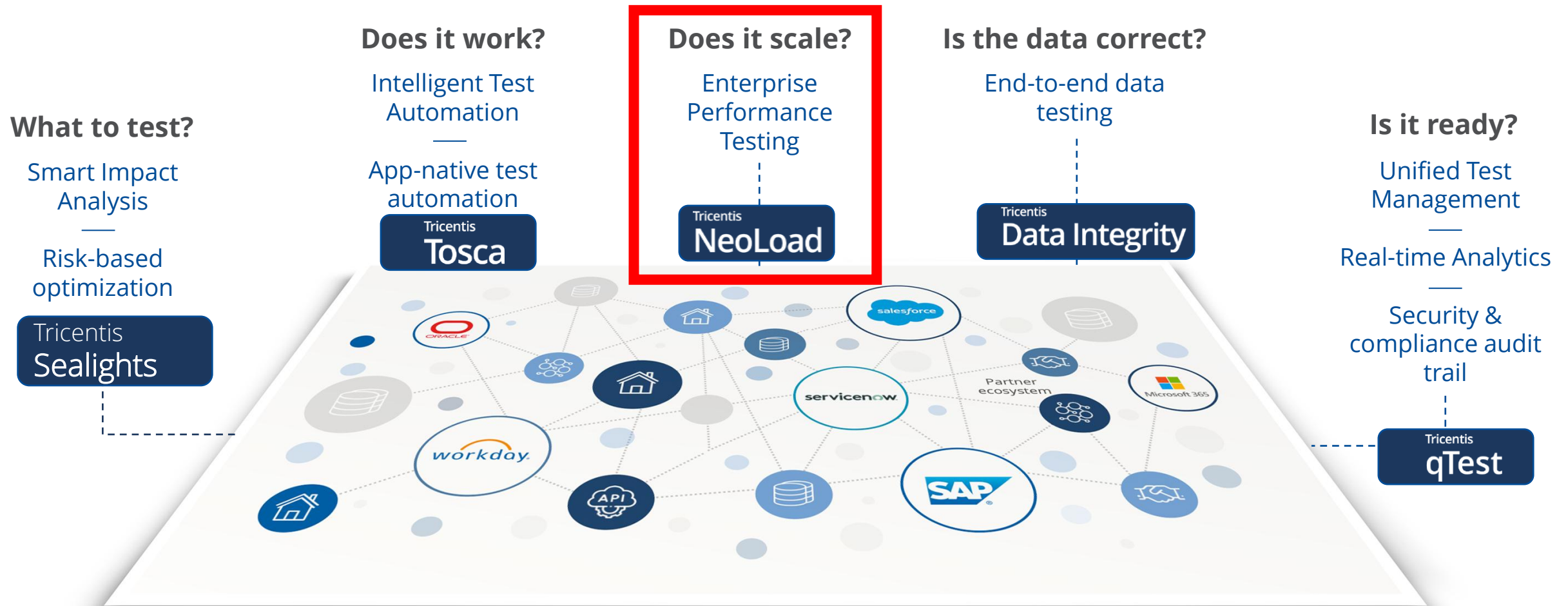
IDC

Named leader in industry analyst vendor ratings

Notre mission

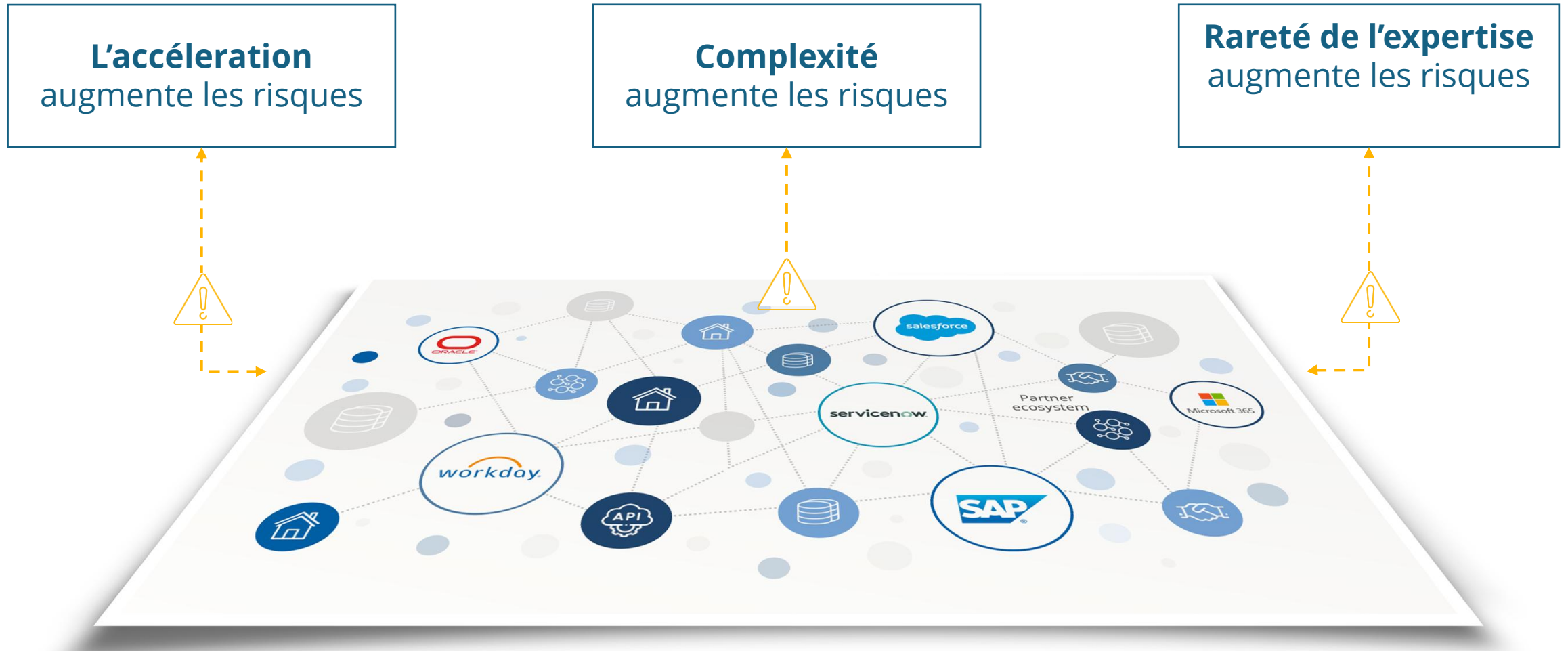
« Permettre à chaque équipe de livrer efficacement des logiciels de haute qualité, rapidement et à grande échelle, pour de meilleurs résultats opérationnels et commerciaux »

La puissance d'une plateforme intégrée



Il n'y a jamais eu autant de risques

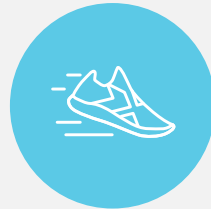
Les approches traditionnelles et en silo ne permettent plus de répondre efficacement



Les chiffres ne mentent pas

L'accélération n'en est qu'à ses débuts

Le Testing ne
peut pas être le
facteur de
ralentissement



1,466%

Augmentation de la
fréquence des MEP
2014-2023

Haute Vitesse



~10x

Augmentation de
la production de
d'ici à 2030 code
avec la **GenAI**

40%

du code
provenant de la
GenAI nécessite
des corrections
massives

L'IA à l'échelle
industrielle

Make Test as
you build, not
only as you
finish a
reality...

Le Shift Left sans oublier le Shift Right



System centric
focused testing



User experience
focused testing





NeoLoad “Next-gen”

Une nouvelle plateforme ouvrant la voie de
l’innovation future



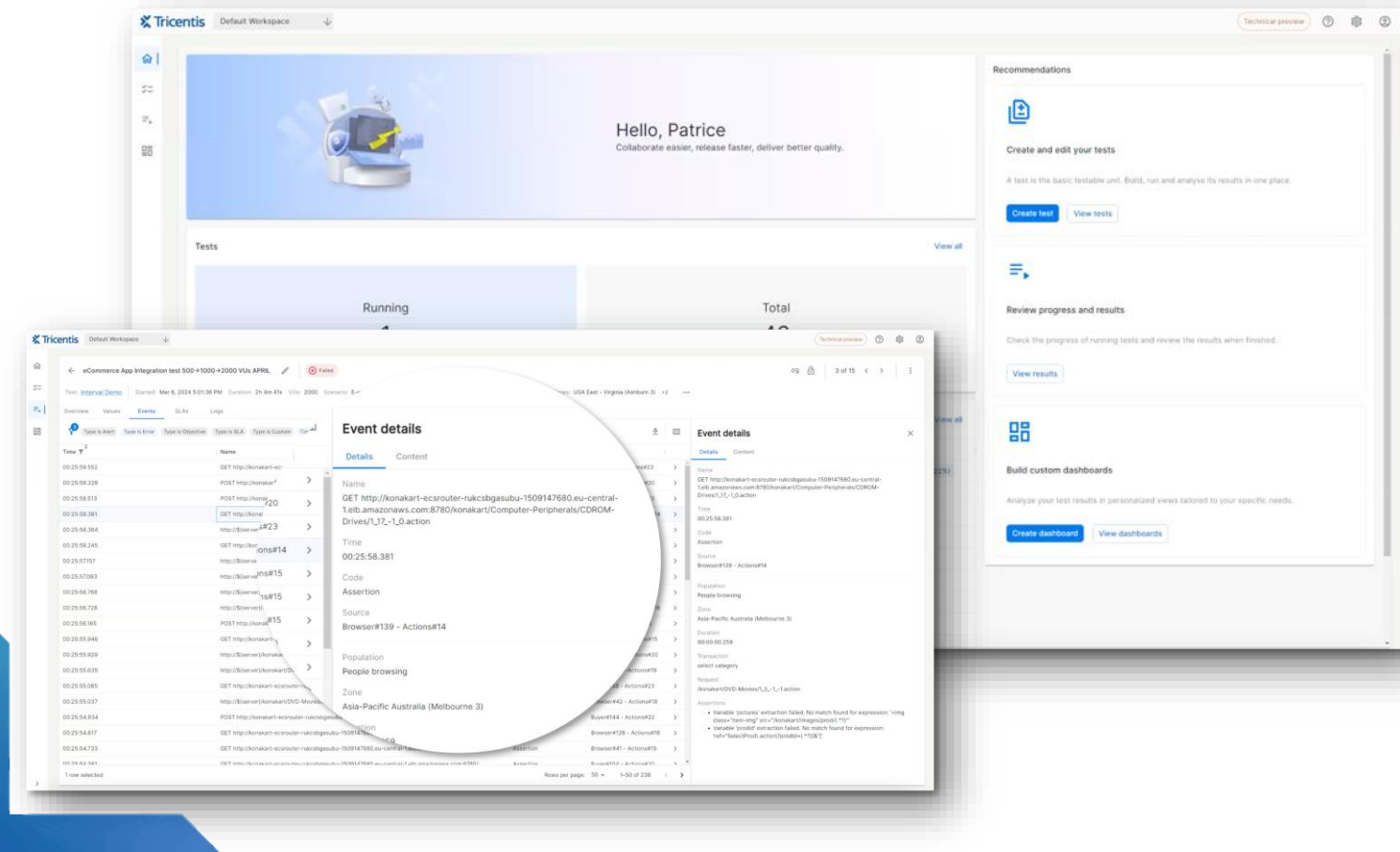
La “Next-gen” de NeoLoad Web

What's new?

A New UI delivering improvements for both usability and to accelerate analysis.

Benefits

- Improved result visibility, customizable dashboards, updated results page
- Enhanced productivity via streamlined navigation
- Easier analysis with deeper filtering
- A new backend architecture
- Access to Augmented Analysis



Benefices

- Extensibilité grâce à une couche API entièrement ouverte
- Permet d'utiliser l'IA et les MCP
- intégrations simplifiées avec les pipelines CI/CD
- Custom connectors pour les toolchains DevOps
- Pret pour le futur et "next-gen AI-driven" performance

Nouvelle Open API Architecture dans NeoLoad Web

API Version 3

NeoLoad API 3.0 OAS 3.0

[./v3/swagger.yaml](#)

Workspaces

Tests

Results

Resources

Information

Infrastructure Providers

Administration of users

Deletion Policies

Schemas

API Version 4

NeoLoad Web API 4.0 OAS 4.0

Welcome to the technical preview of our next-generation API!

Please be aware that this version is under active development, and we can't guarantee compatibility with future versions.

We value your feedback, so it's crucial for the improvement of this API.

Exercise caution and refrain from integrating this into critical production environments.

Access

Authenticate

Checkouts	Execute and manage checkout from an already configured SCM repository
Test Executions	Execute a test and manage it
Dashboard Series	
Dashboard Tiles	
Dashboards	
Dynamic Infrastructure	
Me	
Reservations	
Result Elements	
Result Events	
Result Intervals	
Result Logs	
Result Monitors	
Result SLAs	
Result Statistics	
Result Time Series	
Result Interval generation	
Results	
Scm repositories	
Sessions	
Settings	
Subscriptions	
Tests	
Tests Trends	
User Workspaces	
Users	
Workspaces	
Zones	
Schemas	

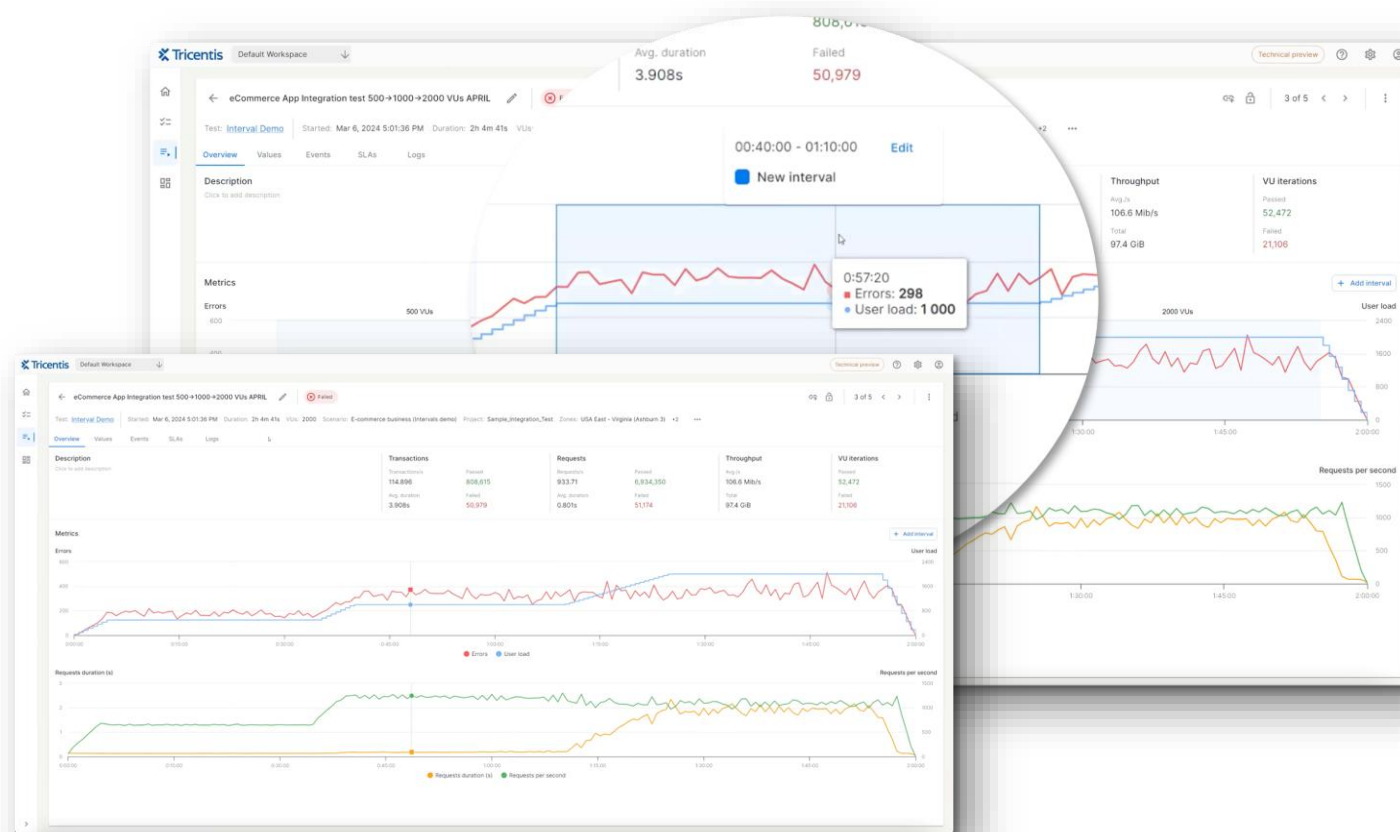
<https://neoload-v2.saas.neotys.com/explore/>

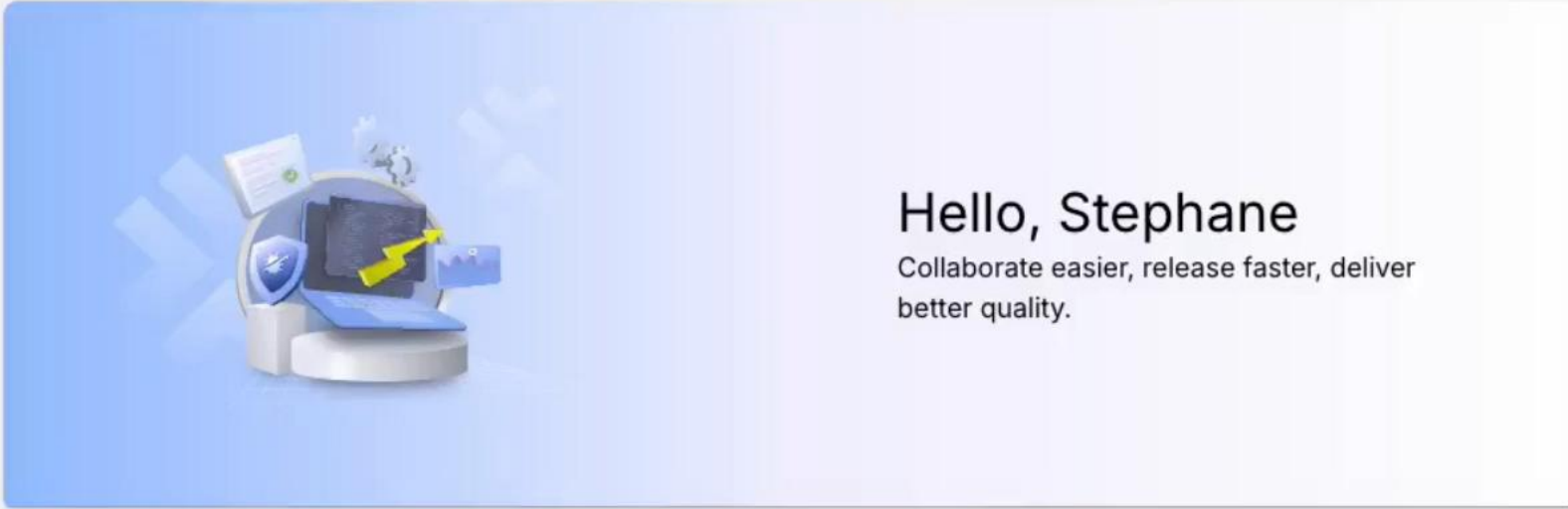
L'analyse augmentée

La première étape de NeoLoad dans l'utilisation d'intelligence artificielle pour simplifier l'analyse des tests de performance

Benefices

- Analyse automatisée “RED” (rate, error, and duration) Identifier rapidement les goulots d'étranglement
- Des conseils et des explications basées sur les résultats
- Simplifier le filtrage, segmentation des tests en intervalles
- Identifier des patterns, trends, et des anomalies



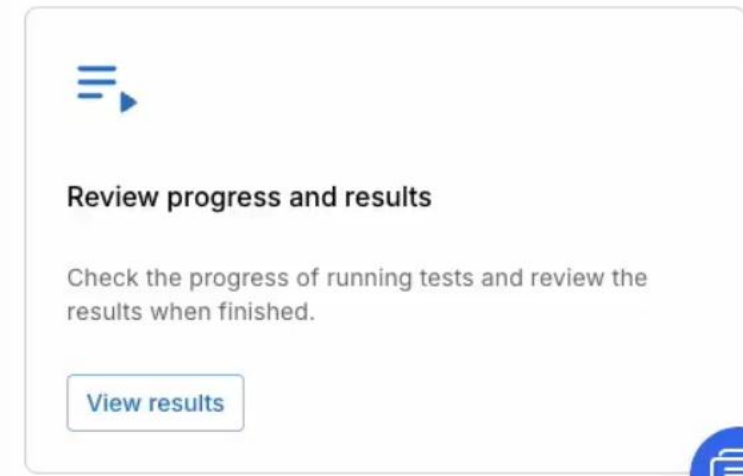


Get it all done in Tricentis NeoLoad



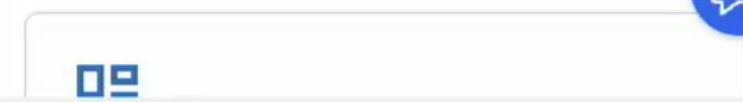
Create and edit your tests

A test is the basic testable unit. Build, run and analyse its results in one place.

[Create test](#)
[View tests](#)


Review progress and results

Check the progress of running tests and review the results when finished.

[View results](#)


Tests

[View all](#)


Latest results

[View all](#)




NeoLoad MCP

Gateway to the AI Ecosystem in Performance Testing

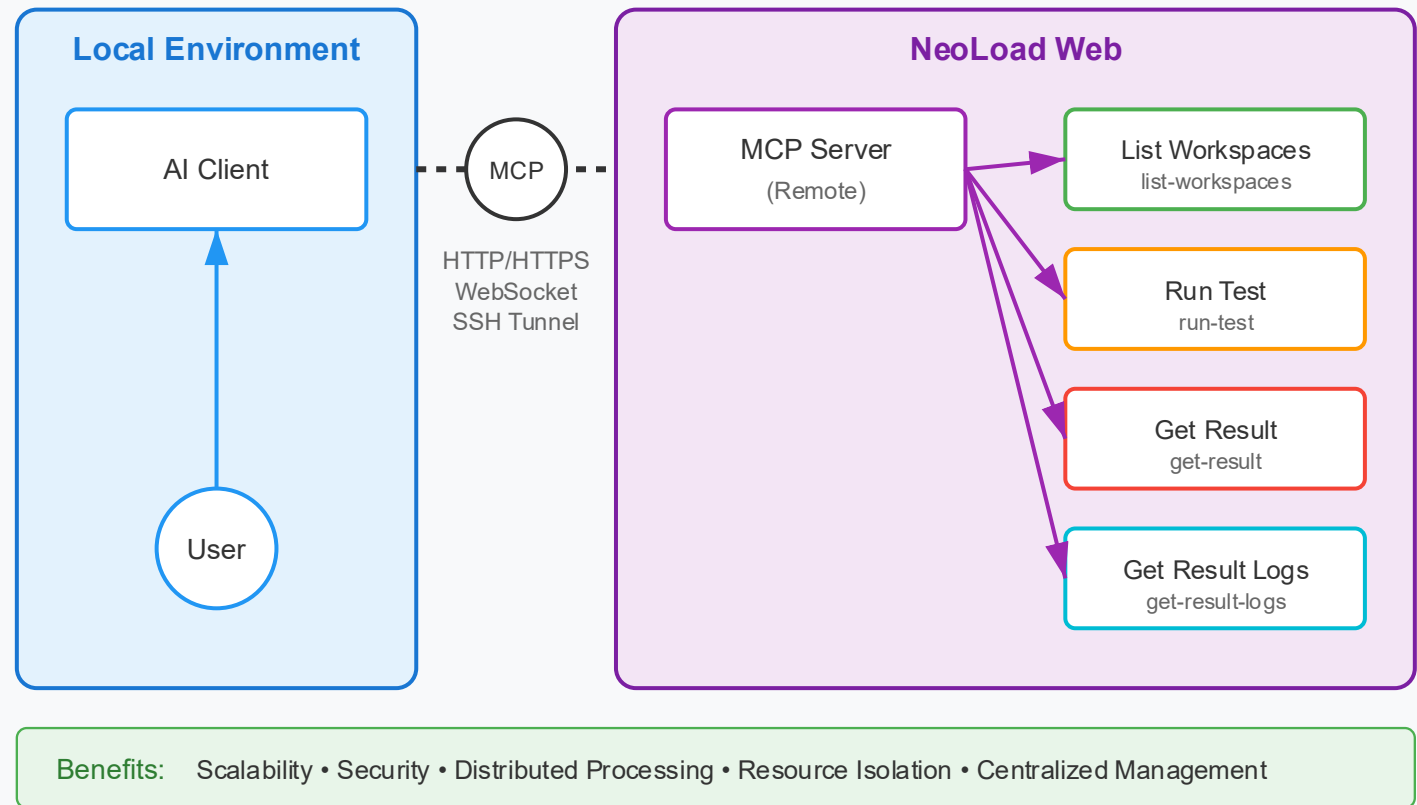


Le remote-MCP NeoLoad

Que veut dire "remote" ?

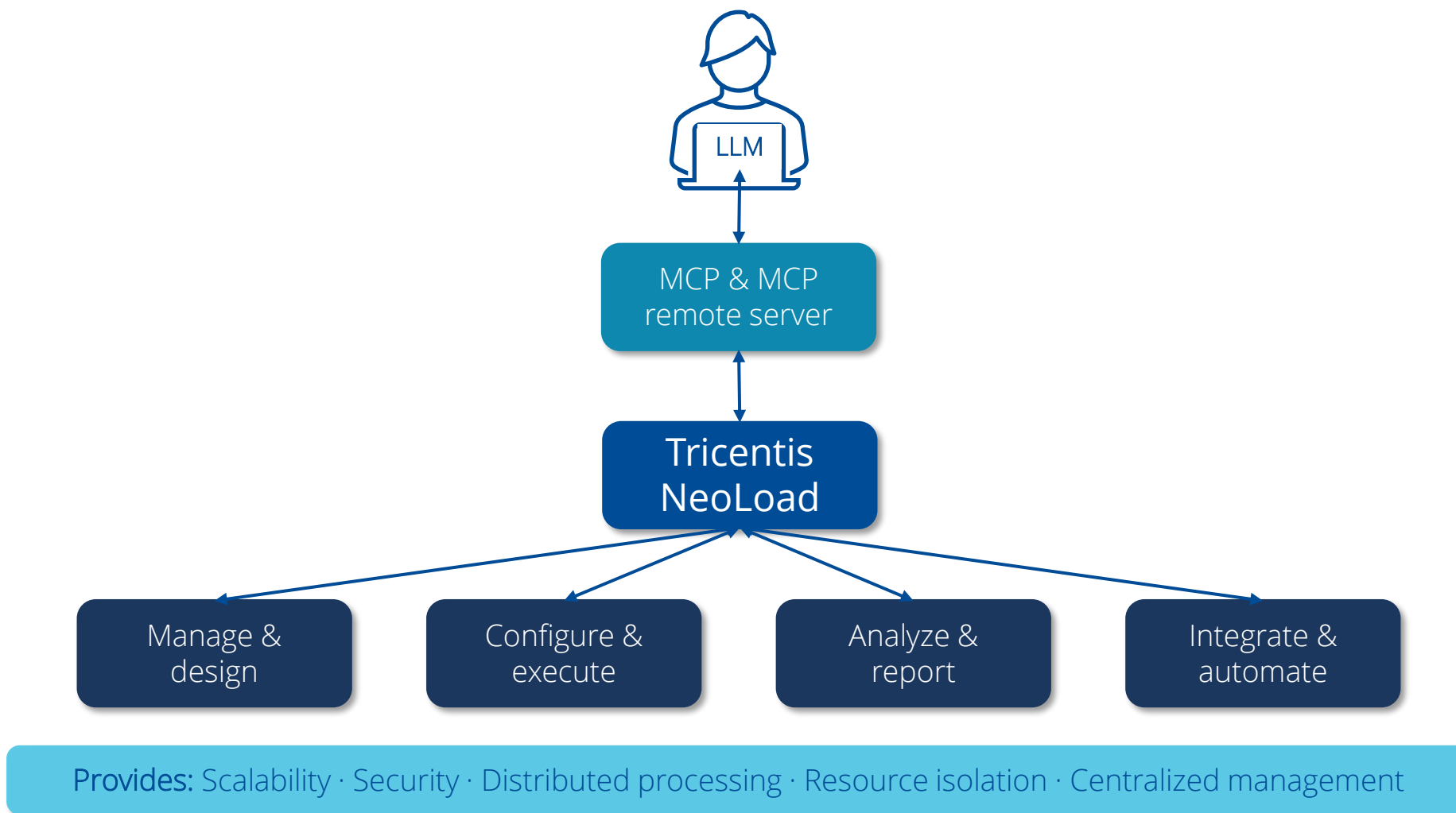
"Remote MCP" permet au Model Context Protocol servers de fonctionner sur des machines ou des environnements différents de ceux du client IA, ce qui rend possible des architectures distribuées et l'exécution d'outils dans le cloud.

Remote MCP Architecture



On fait quoi avec le MCP NeoLoad ?

Une nouvelle manière de s'interfacer pour le performance testing



MCP – Tools

A “**Tool**” est une fonction ou une capacité qui étend les capacités d'un modèle d'IA au-delà de la génération de texte. Les outils permettent au modèle d'effectuer des actions spécifiques.

“Tools” pris en charge par NeoLoad en Juin 2025

- Get-result
- Get-result-events
- Get-result-logs
- List-results-workspace
- List-tests-workspace
- List-workspaces
- Run-test
- Stop-test-execution

Exemples d'utilisation

Tool NeoLoad

Exemple de Prompt

Action

Get-result	Show the summary of the last login performance test.	Le LLM récupère les données détaillées des résultats des tests, notamment le statut, la durée, les SLA et les statistiques d'exécution.
Get-result-events	List any anomalies or threshold breaches from the last test.	Le MCP extrait les événements critiques (par exemple, les violations de SLA, les erreurs) pour un examen en temps réel ou un triage post-test.
Get-results-logs	Identify errors logged during the checkout flow test.	L'assistant récupère les fichiers journaux associés à l'exécution d'un test spécifique afin d'aider à identifier les problèmes.
List-results-workspace	List recent tests run in the performance workspace.	L'agent IA fournit une liste de tous les tests exécutés dans un espace de travail donné à des fins d'examen ou de comparaison.
List-tests-workspace	List available test scenarios in the staging workspace.	Fournit une liste des espaces de travail disponibles, utile pour les utilisateurs qui gèrent ou naviguent entre plusieurs équipes ou environnements.
List-workspaces	List the currently available NeoLoad workspaces.	Fournit une liste des workspaces disponibles, utile pour les utilisateurs qui gèrent ou naviguent entre plusieurs équipes ou environnements.
Run-test	Start the MCP demo test.	Exécute un test défini avec des paramètres spécifiés, déclenché de manière conversationnelle.
Stop-test-execution	Stop the current performance test due to instability.	Interrompt immédiatement un test en cours d'exécution sur la base d'un retour d'information en temps réel ou d'une détection d'anomalies basée sur l'IA.

MCP Tools, en octobre on fait quoi de plus ?

Demandez à Claude et il vous le dira...

Test Management

- **list-workspaces** - List workspaces available to the current user
- **list-tests-workspace** - List available tests in a specific workspace
- **get-test-project** - Retrieve a project from a test by its ID
- **run-test** - Execute a test by its ID with a custom result name
- **stop-test-execution** - Cancel a running test execution

Test Results & Analysis

- **list-results-workspace** - List test results in a specific workspace
- **get-result** - Get detailed test result information by ID
- **get-result-elements** - Retrieve elements from a test result
- **get-result-events** - Get events for a test result with pagination
- **get-result-logs** - Get logs for a test result
- **generate-intervals** - Generate intervals for analysis of a test result
- **list-intervals** - List all intervals for a specific test result

Performance Metrics

- **get-result-requests-values** - Retrieve request performance values (duration percentiles, error rates, throughput)
- **get-result-transactions-values** - Retrieve transaction performance values with similar metrics

Test Configuration

- **get-test-scenario-populations** - Get population settings from a test scenario
- **update-test-scenario-populations** - Update load patterns (constant, ramp-up, peaks, custom)
- **update-selected-test-scenario** - Change the selected scenario for a test

Infrastructure & Zones

- **list-zones** - List available zones (static, dynamic, cloud)
- **create-static-zone** - Create a new static zone
- **create-dynamic-zone** - Create a dynamic zone with specified provider and sizing
- **enable-cloud-zone** - Enable cloud zones in specific regions
- **disable-cloud-zone** - Disable cloud zones
- **delete-static-dynamic-zone** - Delete zones
- **rename-static-zone** - Rename static zones
- **update-dynamic-zone-resource-sizing** - Modify resource allocation for dynamic zones

Infrastructure Providers

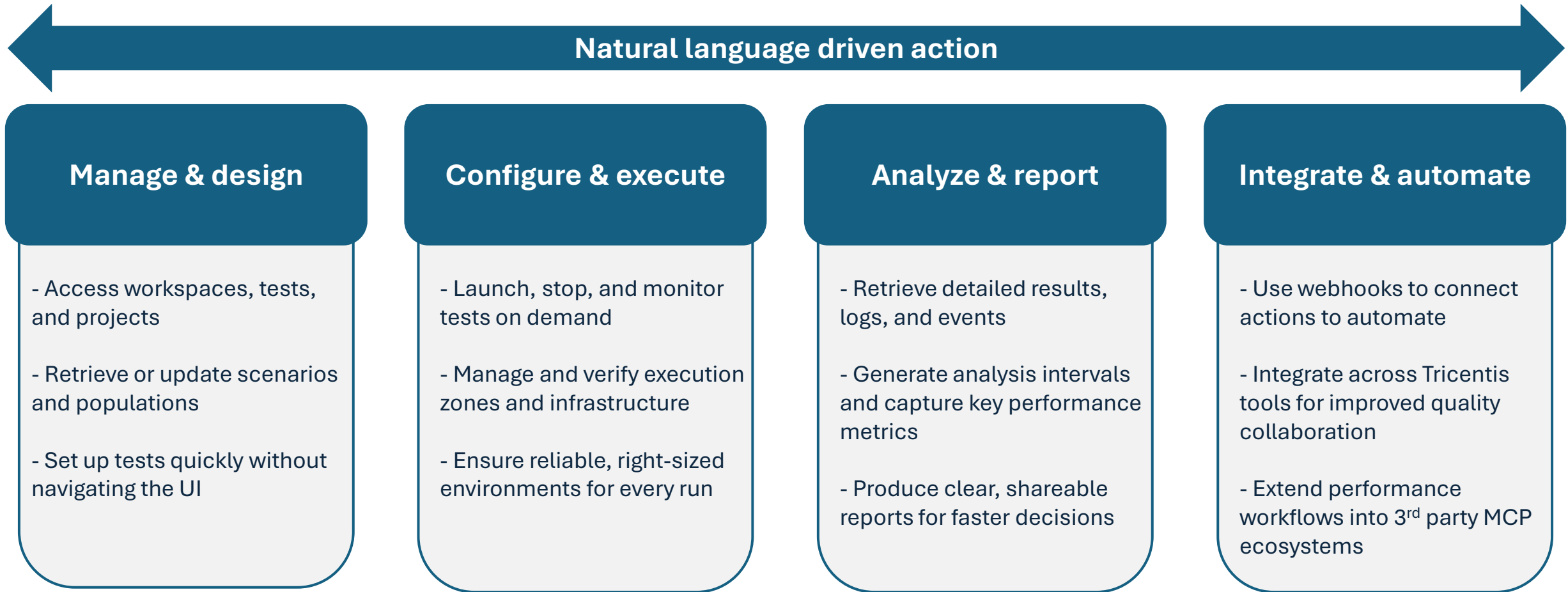
- **list-infrastructure-providers** - List available dynamic zone providers
- **check-infra-provider-availability** - Test connectivity to infrastructure providers

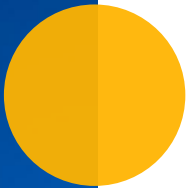
Webhooks

- **get-webhooks** - Retrieve webhooks associated with a workspace

Et concrètement on peut faire quoi avec le MCP ?

Ce n'est pas à propos des tools, mais ce que vous ferez avec...





Un exemple
réel de ce que
l'on peut faire
avec NeoLoad
et l'IA.



Analysis:

For test Flask App Comprehensive Load Test in workspace MCP-AI-Workspace , please analyze the most recent test result using this structured workflow:

Step 1: Error Analysis

- Check system logs for ERROR/WARNING level entries and application failures
- Review test events for failed objectives, SLA breaches, or error rates >0%
- If errors found, investigate root causes using available diagnostic data

Step 2: Performance Element Analysis

- Retrieve all request and transaction performance data
- Rank by average response time to identify top 3 slowest actions/transactions
- Include: avg/min/max duration, throughput, error count, and percentile data

Step 3: Interval Analysis & Issues Detection

- Generate AI-powered intervals if available
- Flag performance issues where: response times increase >20%, throughput drops >15%, or error rates spike
- Document any ramp-up vs plateau performance degradation patterns

Step 4: Comparative Analysis

- Compare the top 3 slowest actions across different time intervals
- Calculate performance variance between intervals for these actions
- Identify which intervals show the worst performance for each action

Step 5: Stakeholder Document

Create a professional report including:

- Executive summary with pass/fail status and key metrics
- Root cause analysis with specific recommendations
- Performance trends and bottleneck identification
- Action items with priorities and success metrics
- Technical appendix with detailed findings

Use clear headings, data tables, and visual indicators (✅❌⚠️) for stakeholder readability.

Claude

File

Edit

View

Window

Help

Flask app load test scenario

List Tests in a Workspace

Now let me get the test project to see the available populations:

Get a Project from a Test

Perfect! Now let me check the current scenario configuration to understand the populations better:

Get Custom Test Scenario Populations

Excellent! I can see the test already has a custom scenario selected. The current configuration shows that the "WebUsers" population already has a very similar 15-minute load pattern to what you're requesting. Let me update it to match your exact specifications:

Update Custom Test Scenario Populations

Let me try with proper formatting:

Update Custom Test Scenario Populations

Perfect!  I've successfully created/updated the custom load scenario for the "Flask App Comprehensive Load Test" with your exact 15-minute load pattern:

Custom Load Scenario Configuration

Test: Flask App Comprehensive Load Test

Duration: 15 minutes

Population: WebUsers (active), HealthMonitors & APITesters (inactive)

Load Pattern Breakdown:

Phase	Time Range	Virtual Users	Description
Phase 1	0:00 - 1:00	1 VU	Baseline period
Phase 2	1:00 - 5:00	1 → 10 VU	Linear ramp-up over 4 minutes
Phase 3	5:00 - 10:00	10 VU	Sustained plateau for 5 minutes
Phase 4	10:00 - 15:00	10 → 1 VU	Gradual ramp-down over 5 minutes

Key Configuration Details:

- Total Duration: 15 minutes
- Peak Load: 10 Virtual Users
- Stop Policy: Current iteration (VUs complete their current iteration before stopping)
- VU Distribution: All load on the WebUsers population

The test is now ready to run with this custom scenario. Would you like me to execute the test or make any adjustments to the load pattern?



Retry

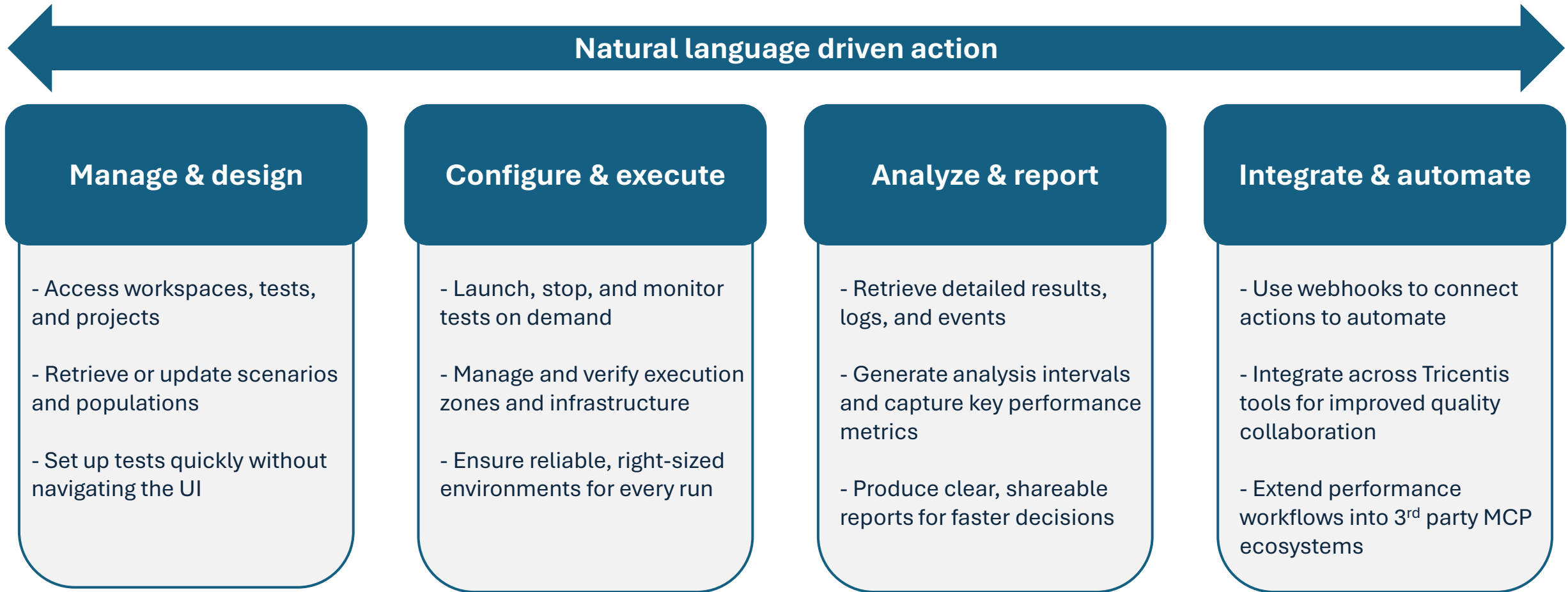
Claude can make mistakes. Please double-check responses.

Reply to Claude...

Sonnet 4.5

Mais encore ?

Ce n'est pas à propos des tools, mais ce que vous ferez avec...





Tridentis C4E

Hi Twan, how are you?

How can I help you today?



Sonnet 4.5



- Write
- Strategize
- Career chat
- Claude's choice

NeoLoad et l'IA: améliorer les capacités des équipes

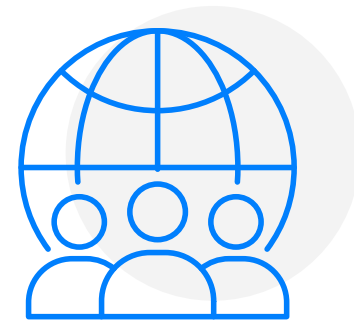
Un effet multiplicateur pour répondre aux besoins de vélocité et de qualité dans des environnement toujours plus complexes



**IA pour réduire les
risques**



**IA pour augmenter la
productivité**



**IA pour augmenter les
compétences**

Questions Answers⁺

8ème édition de la
Soirée du Test
Logiciel
Sophia -
Antipolis



13 octobre 2025
15h à 22h30



Amadeus, Biot



Soirée du
test logiciel

☒ Sophia Antipolis

SCAN ME



**Merci de
votre écoute !**

Votre avis nous intéresse