



Using **Agentic** solutions with IaC for Managing infrastructure “An IoA story”

Peter Bosch - Distinguished engineer
BRKETI-2286

Mission Statement

Outshift by Cisco is the **incubation** engine delivering what's next and new for Cisco: **Emerging** technologies that target **adjacent** markets and **personas** to build **meaningful** businesses and achieve innovative results.

- 1 Power (**adoption**) metrics
- 2 **End-to-End**, solving for more than tech

Innovation, full speed ahead

At Outshift, we turn ideas into action, breaking new ground in **agentic AI**, **quantum**, **next-gen infrastructure**, and more.

What is the Internet of Agents

Internet of
Agents



Agents that can pursue complex goals with **limited direct supervision**, capable of a wide range of actions. And reliable enough that a **user could trust them to act autonomously** on their behalf.

The Internet of Agents is an **open¹**, interoperable, Internet for agent-agent collaboration

Vision Statement for the Internet of Agents

We envision a future where Agentic AI systems transform industries by enabling the developer community, partners, and enterprises to build intelligent, autonomous AI applications.

Enterprises must create agentic workflows and applications **combining internal and 3rd party agents** to fully leverage the power of AI and accelerate their business and unlock new step-change in productivity

Through **open**, interoperable standards and a **multi-vendor** agentic ecosystem, the collective enables innovation, fosters collaboration, and unlocks new growth opportunities.

An open, neutral and interoperable Internet of Agents drives **maximum value** for all players: infra and tool builders, operators, app developers, and consumers. It avoids vendor lock-in.

Goal: Empower the collective to build the Internet of Agents

Why is **agentic** interesting?

It is a business opportunity ... with society impacts

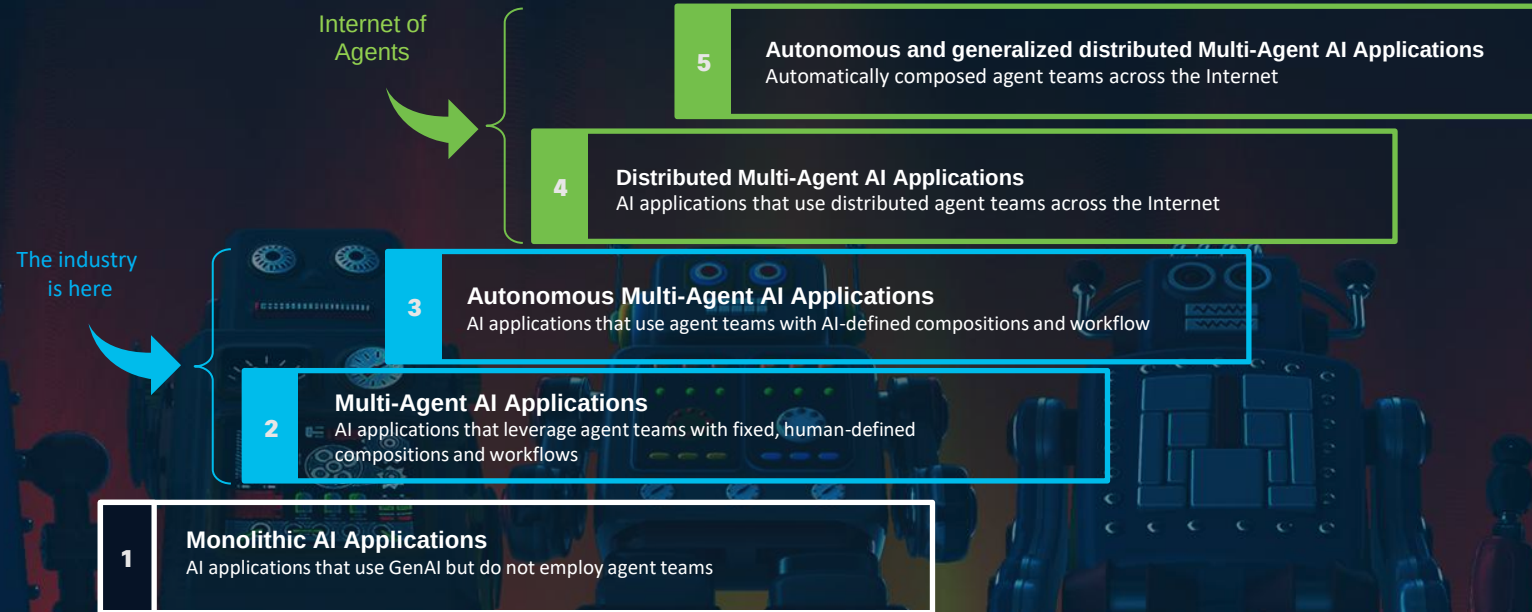
“... a key insight of Autor, Levy, and Murnane (2003) is that an occupation can be viewed as a bundle of tasks, some of which offer better applications for technology than others ...”

“... In the case of ML, we find that (i) most occupations in most industries have at least some tasks that are *[Suitable for Machine Learning]* SML; (ii) few if any occupations have all tasks that are SML; and (iii) unleashing ML potential will require significant redesign of the task content of jobs, as SML and non-SML tasks within occupations are unbundled and rebundled. ...”

What Can Machines Learn and What Does It Mean for Occupations and the Economy?† By Erik Brynjolfsson, Tom Mitchell, and Daniel Rock*

The evolution of AI systems

The start of the Internet of Agents



What capabilities does the **IoA** need to provide¹

Open source code for tool and service

Agent Identity, Announcement, Discovery and Verification

Agent to agent connectivity and interworking

Agentic ensemble workflows, composition, deployment and supervision²

Interoperability between agents from multiple agentic frameworks and contexts

Agentic ensemble observability, evaluation, metering

Agentic ensemble trust and security

This is a distributed systems problem

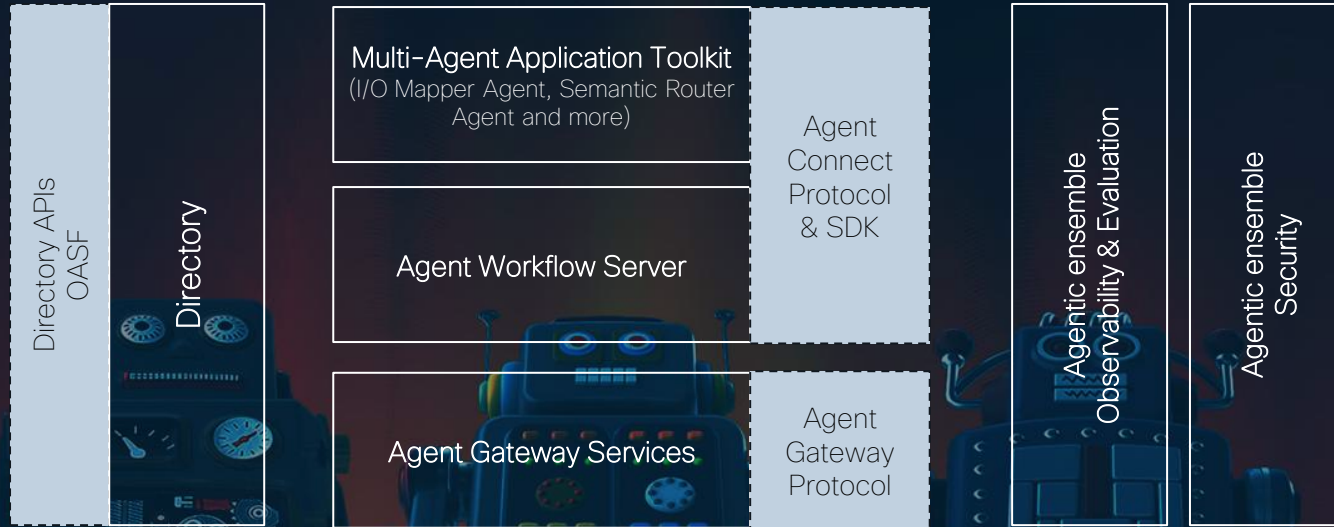
^[1] These capabilities will be available progressively – the collective will prioritize the IoA roadmap

^[2] IoA is optimized to take full advantage of the cloud native stack, allowing scalable and high-availability multi-agent application deployments

Internet of Agents architecture



Internet of Agents architecture



Challenges in Human-Agent Collaboration for the Internet of Agents

Gaps in Contextual Understanding

Sharing More Than Data:

AI agents often share raw data instead of complex states, limiting meaningful collaboration.

Missed Human Nuances:

AI systems fail to align with team dynamics, intuition, and shared understanding.

Probabilistic Operations

Inconsistent Outcomes:

Probabilistic operations lead to variable and unpredictable results that disrupt workflows.

Misinterpretation of Intent:

AI agents struggle to fully grasp human intent, resulting in decisions that deviate from team priorities.

Challenges in Mental Models

Hard to Keep Up:

Evolving AI systems make it difficult for humans to maintain accurate mental models.

Divergent Operational Logic:

AI systems' behavior often doesn't align with human mental models, creating frustration and inefficiency.

Trust Over Time

Initial Overconfidence:

Users often overestimate AI's capabilities, but early errors reveal limitations, causing trust to decline.

Trust Through Reliability:

Long-term trust builds as AI proves consistent, transparent, and aligned with user goals.

Internet of Agents **sample applications** and tooling

SRE and IT tasks

SRE is usually overwhelmed with “running the infrastructure”: Too many jobs, too few people, not enough time to do it right.

Build out a set of agentic functions to add the SRE person managing and controlling the cloud infrastructure

Network configuration

Managing networks is a tedious task. Are all ports well connected? Do I waste IP addresses? Are there firewall leaks? Are all certs up to speed?

Let agents do the “boring” stuff. Machines are much better and cheaper at such tedious jobs.

Learn-on-the-job

Many tasks are difficult to formalize as these grew organically. Human resources, customer management, PM/PO, infra management, campaigns, and more.

Make the agents learn on the job by observation and repetition to help humans be more efficient.

Distributed tooling

To make distributed agentic applications work, a lot of tooling is needed to harmonize between agentic systems, agents, humans, tools and dynamic data.

Build out the IoA collective to realize this across the industry. It needs to be an “OpenAPI-like” effort

IoA and TF/GH/PRs

(Or: Internet of Agents and
Terraform/Github/Pull-requests)

Can AI help set up my infrastructure?

Infrastructure management is programming

Infrastructure providers can be managed through API and intent-based configuration is essential

Intent as Code or as Data

Infrastructure can be described as Code or as Data. Can we find a general AI-based solution addressing both approaches?

Terraform is a programming language (IaC)

Like any program, how does one know that the application did its job well? How can you get insight in the side effects? How do you debug?

Can AI provide for deep insights in the “app”?

AI-based tools provide feedback on the program and real state, but can we trust the results? Hallucinations, opinions, and bad reasoning etc

Can agents provide better results?

Collaboration between agents with multiple models can provide better insights and verify each other. How far can we go?

What is Terraform?

Terraform is an infrastructure as code tool that lets you build, change, and version cloud and on-prem resources safely and efficiently.

(<https://developer.hashicorp.com/terraform/intro>)



Use Cases: managing cloud, networks, apps

- Terraform best practices
- Code modularity, reuse & consistency
- Policy & Compliance
- Provider versioning
- Security
- Cost
- Documentation

Provision

[using IaC / GitOps]



Test

- Performance
- Load
- Security
- Latency

Plan

- Capacity planning
- Resource optimization
- Migrations
- Automation improvement
- New tools or services

Monitor

- Drift
- Observability
- Incident response

A Terraform example

```
# Define the AWS provider
provider "aws" {
  region = "us-east-1" # Change to your preferred region
}

# Create a new EC2 instance
resource "aws_instance" "web" {
  ami      = "ami-0c55b159cbf9e1f0" # Amazon Linux 2 AMI (Replace
with your preferred AMI)
  instance_type = "t2.micro"

  tags = {
    Name = "MyTerraformInstance"
  }

  # Output the public IP of the instance
  output "instance_ip" {
    value = aws_instance.web.public_ip
  }
}
```

(per GPT)

A Terraform example

```
# Define the AWS provider
provider "aws" {
  region = "us-east-1" # Change to your preferred region
}

# Create a new EC2 instance
resource "aws_instance" "web" {
  ami      = "ami-0c55b159cbf9e1f0" # Amazon Linux 2 AMI (Replace
with your preferred AMI)
  instance_type = "t2.xlarge"

  tags = {
    Name = "MyTerraformInstance"
  }

  # Output the public IP of the instance
  output "instance_ip" {
    value = aws_instance.web.public_ip
  }
}
```

(per GPT)

What are the consequences?

A GitHub/Terraform change example

```
diff --git a/main.tf b/main.tf
index 6d875b1..3ab7686 100644
--- a/main.tf
+++ b/main.tf
@@ -5,14 +5,14 @@ provider "aws" {
 # Create a VPC
 resource "aws_vpc" "main_vpc" {
- cidr_block = "10.1.0.0/16"
+ cidr_block = "10.0.0.0/16"
 }
```

```

-# Create public and private subnets
+# Create a public subnet
resource "aws_subnet" "public_subnet" {
  vpc_id      = aws_vpc.main_vpc.id
  cidr_block  = "10.0.1.0/24"
@@ -23,6 +23,7 @@ resource "aws_subnet" "public_subnet" {
}

+# Create a private subnet
resource "aws_subnet" "private_subnet" {
  vpc_id      = aws_vpc.main_vpc.id
  cidr_block  = "10.1.2.0/24"
@@ -62,7 +63,7 @@ resource "aws_route_table_association" "public_rt_assoc" {
  route_table_id = aws_route_table.public_rt.id
}
```

```

-# Security group allowing SSH and HTTP access
+# Create a security group allowing SSH and HTTP access
resource "aws_security_group" "main_sg" {
  vpc_id = aws_vpc.main_vpc.id
@@ -94,17 +95,62 @@ resource "aws_security_group" "main_sg" {
 # Create an EC2 instance
resource "aws_instance" "web_server" {
- ami      = "ami-0c55b159cbfafa1f0" # Example AMI
+ ami      = "ami-0a897ba00eae7398" # Amazon Linux 2023 AMI
  instance_type = "t2.micro"
- subnet_id = aws_subnet.public_subnet.id
- security_groups = [aws_security_group.main_sg.name]
+ private_ip = "10.1.2.100"
+ subnet_id = aws_subnet.private_subnet.id
+
+ security_groups = [aws_security_group.main_sg.id]
}

Et cetera.
```

What can we say about these changes?

GitHub/Terraform problems

Multi-repo updates are not correlated
Multiple changes in single PRs confuse tools
No correlation to company policies and regulations
No insight in unwanted side effects

Unhelpful tools to assess the impact of changes
Generalized feedback instead of specialized feedback

Understanding the PR - a taxonomy of approaches



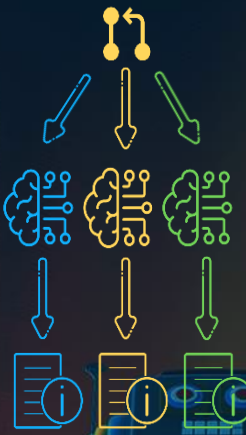
Deterministic answers with existing analysers

Assess the quality of the TF using well-defined rules, plus **lots trial & error**



Probabilistic answers with trained model

Today's **pre-trained models** offer decent insights but need massive prompts to prevent hallucinations and get consistency. **Brittle.**



Multiple probabilistic answers

Smaller, tuned, task-specific agents give better **consistent answers** with small prompts. Diverse viewpoints overlap, contradict and offer partial answers



Harmonise the answers into a common view

Remove the confusion by providing a **common and harmonised answer**



Iteratively improve on the answers

Feedback the outcome across the team members for a **better overall answer**

Understanding the PR with a single prompt

If some lines are indented more than others, dedent can't normalize it effectively.

```
system_message = wrap_prompt("""
```

You are a senior software engineer, specialized in IaC, tasked with reviewing code changes in a pull request.

You will get a GitHub pull request which shows all the added and deleted lines, just like how GitHub shows it on their UI.

Your task is to review the modifications and provide feedback on them, using the same language and logic as teammates would do when reviewing a PR.

Input from the user:

- You will receive all the Terraform files from the user in the "FILES" list.

- You will receive the changes which were done by the user in another array. These are the changes which were done compared to the target branch, to reach the current state of the files.

[...]

Your output format:

- Output MUST be in JSON, with the following instructions:

- Each comment has to belong to a change object from the changes list.

[...]

- DO NOT USE markdown in the response.

Focus your review on the following areas:

- Code Quality: Ensure that the code follows best practices for readability, maintainability, and clarity.

- Terraform Best Practices: Review the Terraform code for adherence to best practices, including proper resource naming, proper use of modules, and idempotency.

[...]

Review Guidelines:

- Review all the files to understand the current state of the codebase.

- Review the changes to understand what was changed in this PR to arrive at the current state of the files.

- Add your comments to the changes and only to the changes.

[...]

Before returning your response, take your time to review your results:

- Make sure that each comment belongs to a change.

[...]

```
""")
```



Single prompt answers

outputs.tf

```
6 - output "public_subnet_id" {  
7   description = "The ID of the public subnet"  
8   value       = aws_subnet.public_subnet.id  
6 + output "password" {
```



alfred-dev-local (bot) on Nov 19, 2024

...

Output value should not be hardcoded.



Single prompt answers

```
7 - description = "The ID of the public subnet"
8 - value       = aws_subnet.public_subnet.id
6 + output "password" {
7 +   description = "The password"
```



alfred-dev-local (bot) on Nov 20, 2024

...

Hardcoded sensitive information (password) in code can lead to security vulnerabilities.



Single prompt answers

```
6 - output "public_subnet_id" {  
7   description = "The ID of the public subnet"  
8   value       = aws_subnet.public_subnet.id  
6 + output "password" {
```



alfred-dev-local bot on Nov 20, 2024

...

Invalid value type for output 'password'. The value should be a reference to a resource attribute.



Single prompt answers

```
6 - output "public_subnet_id" {  
7   -   description = "The ID of the public subnet"  
8   -   value       = aws_subnet.public_subnet.id  
6 + output "password" {
```



alfred-dev-local (bot) on Nov 20, 2024



Output name 'password' is not descriptive and should not be hardcoded. Avoid using sensitive data in outputs.



Single prompt answers

```
7 - description = "The ID of the public subnet"
8 - value       = aws_subnet.public_subnet.id
6 + output "password" {
7 +   description = "The password"
```



alfred-dev-local bot on Nov 19, 2024



The output variable name 'password' indicates that a sensitive value is being exposed. This can lead to unintended information disclosure.



Single prompt answers

```
7 - description = "The ID of the public subnet"
8 - value       = aws_subnet.public_subnet.id
6 + output "password" {
7 +   description = "The password"
```



alfred-dev-local bot on Nov 19, 2024



Sensitive information such as passwords should not be hardcoded in the code. This exposes the password in plaintext, which is a significant security risk.



Single prompt answers

```
6 - output "public_subnet_id" {  
7   - description = "The ID of the public subnet"  
8   - value       = aws_subnet.public_subnet.id  
6 + output "password" {
```



alfred-dev-local bot on Nov 19, 2024



Use of sensitive data in the 'password' output.



Single prompt answers

```
6 - output "public_subnet_id" {  
7   - description = "The ID of the public subnet"  
8   - value       = aws_subnet.public_subnet.id  
6 + output "password" {
```



alfred-dev-local bot on Nov 19, 2024



Output name 'password' is not descriptive.



Single prompt answers

PR title suggestion: Replace public subnet ID output with password output in outputs.tf

PR description suggestion:

This pull request makes the following changes to the `outputs.tf` file:

- Removes the `public_subnet_id` output block, which included:
 - description: "The ID of the public subnet"
 - value: `aws_subnet.public_subnet.id`
- Adds a new `password` output block, which includes:
 - description: "The password"
 - value: `1234Password`

These changes will replace the public subnet ID output with a password output in the Terraform configuration.

Single prompt answers

PR title suggestion:

Update output.tf to replace public_subnet_id with password output

PR description suggestion:

This pull request makes the following changes to the `outputs.tf` file:

- Removes the `public_subnet_id` output, which included the description "The ID of the public subnet" and the value `aws_subnet.public_subnet.id`.
- Adds a new `password` output, which includes the description "The password" and the value `1234Password`.

These changes update the outputs to include a password instead of the public subnet ID.

Single prompt answers

*A prompt of 7KB!
Programming by instruction – sort of*

Unwieldy and imprecise!

*Inconsistent results, repetitions and
unhelpful*

Hallucinations

A smarter approach

"Agentic Map-Reduce"

```
# Define the AWS provider
provider "aws" {
  region = "us-east-1" # Change to your preferred region
}

# Create a new EC2 instance
resource "aws_instance" "web" {
  ami      = "ami-0c55b159cbf1f0" # Amazon Linux 2
  AMI (Replace with your preferred AMI)
  instance_type = "t2.micro"

  tags = {
    Name = "MyTerraformInstance"
  }
}

# Output the public IP of the instance
output "instance_ip" {
  value = aws_instance.web.public_ip
}
```

Take the Terraform-based app and PR (for specific queries) ...

.. decompose into smaller trees ...

... process by many agents ...

... and tune towards a better result (automated or manual)

... into the overall view ...

... combine the results ...

Decomposing the problem

prompts='Explain the meaning of the included TF file; cite Terraform resources; expand resource names to fully qualified resource names.'

prompt='Correlate variables by their fully qualified resource names between the various outputs; present the full name of the variables and the old/new values; report implications of changes'

prompt='What happens when the following diff is applied (- depicts a dropped line, + an added line) to the configuration; present fully qualified resources names of the resources affected, recursively'

prompt='What happens when the following diff is applied (- depicts a dropped line, + an added line); cite Terraform resources; use fully qualified resource names'

Define assistants, tools and agents that are tasked with one thing and one thing only

Combining intermediate results

Model response on ['alfred-test-main/main.tf']: The provided Terraform configuration sets up a basic AWS infrastructure with the following components:

[...]

2. **VPC (Virtual Private Cloud):**

- Creates a VPC named `main\_vpc` with a CIDR block of `10.0.0.0/16`.

3. **Subnets:**

- Creates a `public\_subnet` within the VPC with CIDR block `10.0.1.0/24`, in availability zone `us-west-2a`.
- Creates a **private_subnet** within the VPC with CIDR block `10.0.2.0/24`, also in `us-west-2a`.

[...]

6. **Security Group:**

- Creates a security group `main\_sg` that allows inbound SSH (port 22) and HTTP (port 80) access from anywhere (`0.0.0.0/0`) and allows all outbound traffic.

[...]

Model response on ['alfred-test-main/outputs.tf']: The code you have provided defines Terraform output values that you can use to obtain specific infrastructure details after a Terraform `apply` command is run. Here's a breakdown of what each block does:

1. **Output "vpc_id":**

- **Description:** Gives a brief explanation of the output.
- **Value:** Outputs the ID of the main VPC defined in your infrastructure as `aws\_vpc.main\_vpc.id`.

2. **Output "public_subnet_id":**

- **Description:** Explains that this output represents the ID of the public subnet.
- **Value:** Outputs the ID of the public subnet specified as `aws\_subnet.public\_subnet.id`.

3. **Output "private_subnet_id":**

- **Description:** Indicates this is the ID for a private subnet.
- **Value:** Outputs the ID of the private subnet defined as **aws_subnet.private_subnet.id**.

[...]

Make sure that individual results can be combined

Drawing conclusions from results

Model response on ['names.main', 'pull-request']: When applying the specified diff to the Terraform configuration, here are the fully qualified resource names of the affected resources, recursively:

1. `**`aws_vpc.main_vpc`**` - Directly affected by the change in the ``cidr_block``.
2. `**Dependents of `aws_vpc.main_vpc`**`:
 - `**`aws_subnet.public_subnet`**` - Because it depends on the VPC ID and its CIDR range could potentially overlap or require adjustment due to the VPC CIDR change.
 - `**`aws_subnet.private_subnet`**` - Similarly affected as the public subnet.
 - `**`aws_internet_gateway.igw`**` - As it's linked to the VPC, changes could indirectly affect network connectivity.
 - `**`aws_route_table.public_rt`**` - The association with the VPC means any change to VPC configuration affects it.
 - `**`aws_route_table_association.public_rt_assoc`**` - Relies on the public route table and subnet, both linked to the VPC.
 - `**`aws_security_group.main_sg`**` - Uses the VPC for its network scope.
3. `**Indirectly affected resource due to changes in `aws_subnet.public_subnet`**`:
 - `**`aws_instance.web_server`**` - As it is associated with the ``public_subnet``, changes in the subnet due to VPC CIDR change can impact instance placement.

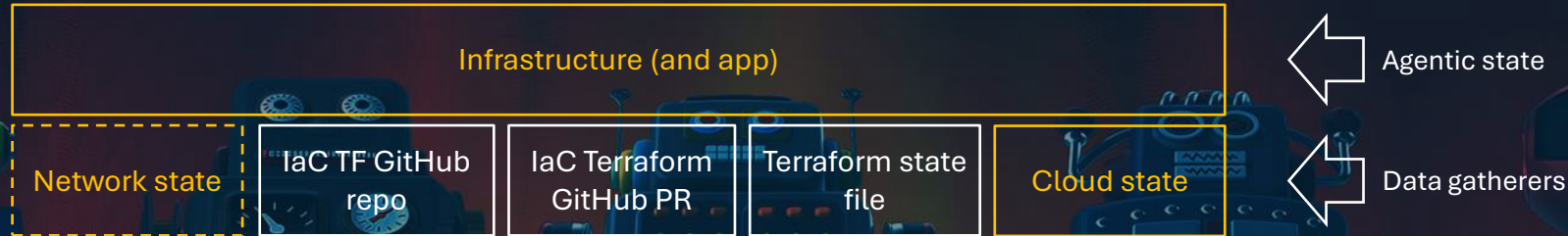
These resources will likely need to re-evaluate or modify configuration due to the adjustment in the VPC's CIDR block.

Simple prompts lead to precise and repeatable results

“Agentic Map-Reduce”

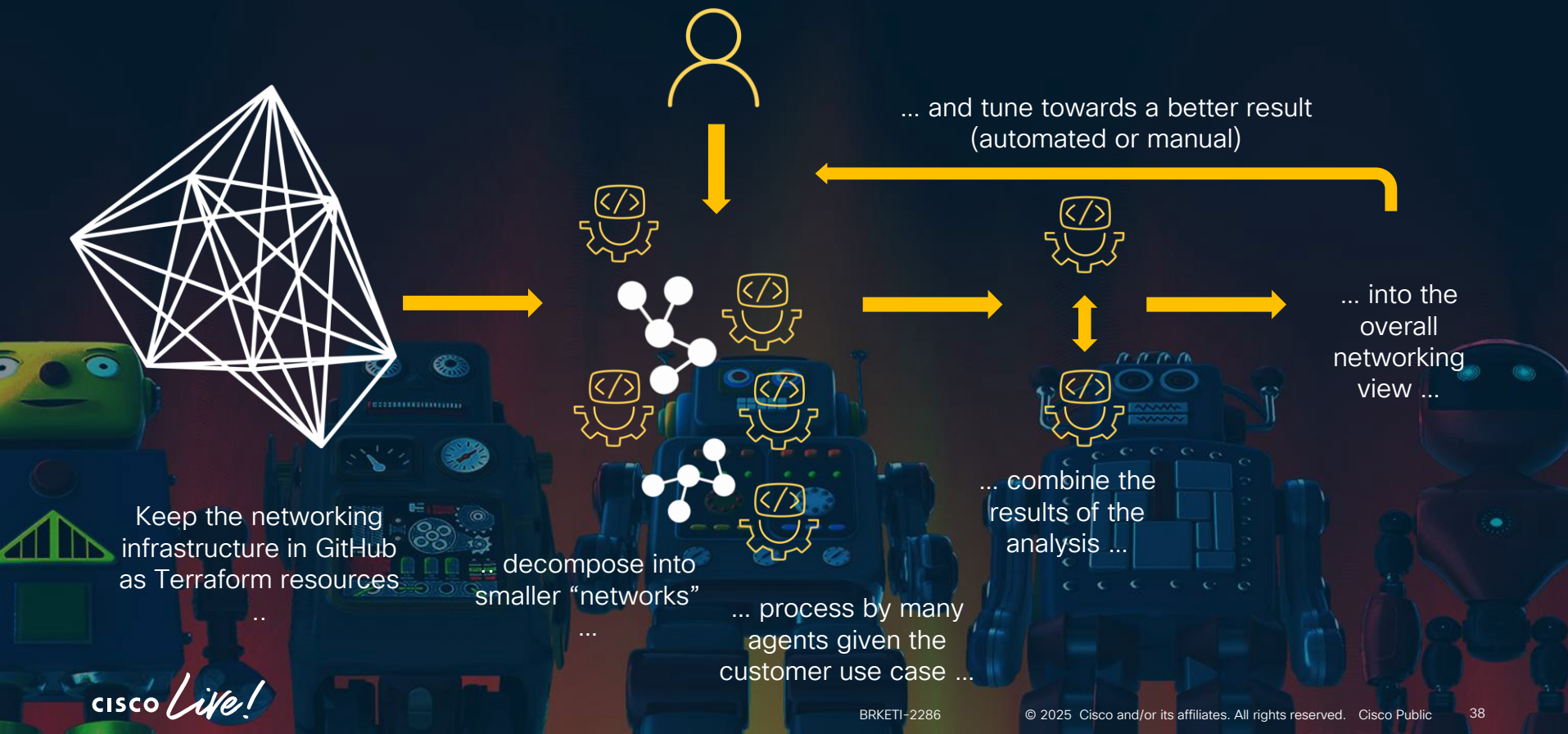
Why github and Terraform?

Use case compositions, comprised of one or more agentic tools



Transactional database for code
Ideal for human/agentic operations

Networking Agentic Map-Reduce



NetDevOps

Network automation use cases

Build and maintain a network-source-of-truth automatically

Manage a network despite it being a moving target with moving parts, vendors, configurations and dinosaurs.

Auto-manage network expansion

Auto-test and enforce compliance and free-format network requirements

Create a single pane of glass for a multi-vendor and diverse network environment

Provide auto-tooling for network problem analysis and remediation

Manage certificate and software loads automatically

Auto-test new or existing deployments before roll-out

“... and potentially run my entire network, I'll tell you where to focus!”

The evolution of the Internet of Agents

Internet of
Agents

The Internet of Agents is coming; agents will collaborate with roles **currently performed by humans**

A simple example of this is **the IaC use case**. Why do still do these tasks manually?



Agentic AI

<u>Date/Time</u>	<u>Session Code</u>	<u>Title</u>	<u>Speaker</u>
Feb 10 / 10:30	DEVLIT-1973	AIPO: A glimpse into the future of GenAI-Powered agile workflows	Peter Bosch
Feb 10 / 16:30	CENETI-1280	Inside Cisco's Incubation Engine: From Bold Bets to Enterprise Impact	Vijoy Pandey
Feb 11 / 12:10	AIHUB-2170	AI Agents and Agentic Frameworks: An Overview	Frank Brockners
Feb 12 / 12:45	CENAI-1310	From infrastructure to AI Agents: Building a framework for trustworthy AI	Vijoy Pandey
Feb 12 / 14:30	BRKETI-2172	A new era of cross-domain diagnostics with AI and Agentic systems	Frank Brockners
Feb 12 / 14:30	DEVNET-2940	Agentic IaC, security, and networking, the good, the bad, and the ugly	Peter Bosch



Next-Gen Infrastructure

<u>Date/Time</u>	<u>Session Code</u>	<u>Title</u>	<u>Speaker</u>
Feb 10 / 10:30	DEVLIT-1973	AIPO: A glimpse into the future of GenAI-Powered agile workflows	Peter Bosch
Feb 11 / 8:00	BRKETI-1005	From Machine Learning to GenAI and Beyond: An Introduction to AI Technologies	Ivan Padilla Ojeda
Feb 12 / 12:45	CENAI-1310	From Infrastructure to AI Agents: Building a framework for trustworthy AI	Vijoy Pandey
Feb 12 / 14:50	AIHUB-2596	How to use multimodal LLMs to unlock enterprises data for key use cases	Mahesh Viswanathan
Feb 13 / 14:00	BRKAI-2552	Multimodality is the future of GenAI	Ramin Pishehvar
Feb 13 / 14:10	AIHUB-2006	Applying AI/ML methods to diagnose network issues using telemetry data	Frank Brockners

The Internet of Agents is here



Want to learn more about
Outshift and Internet of Agents?

Webex App

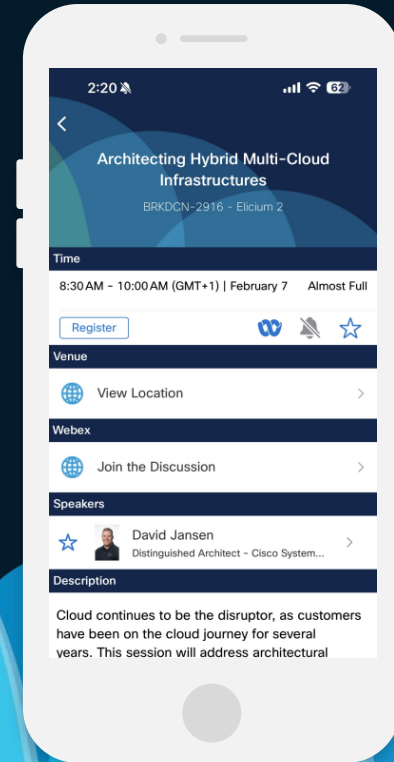
Questions?

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.



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

Continue your education



- 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.

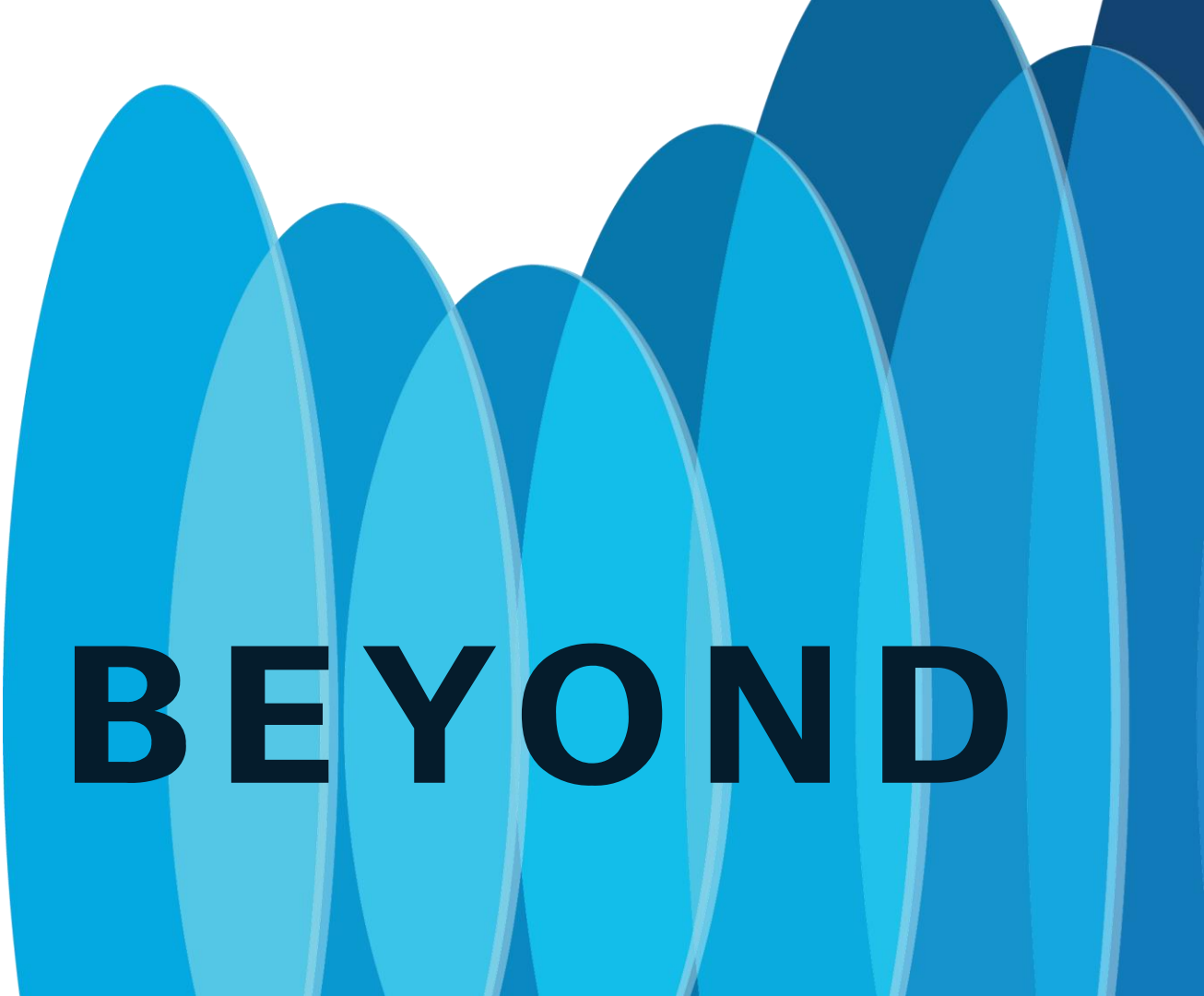


Thank you



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 horizon or a series of waves. The overall aesthetic is clean and modern.