CREST
— Ideate — Innovate — Implement —

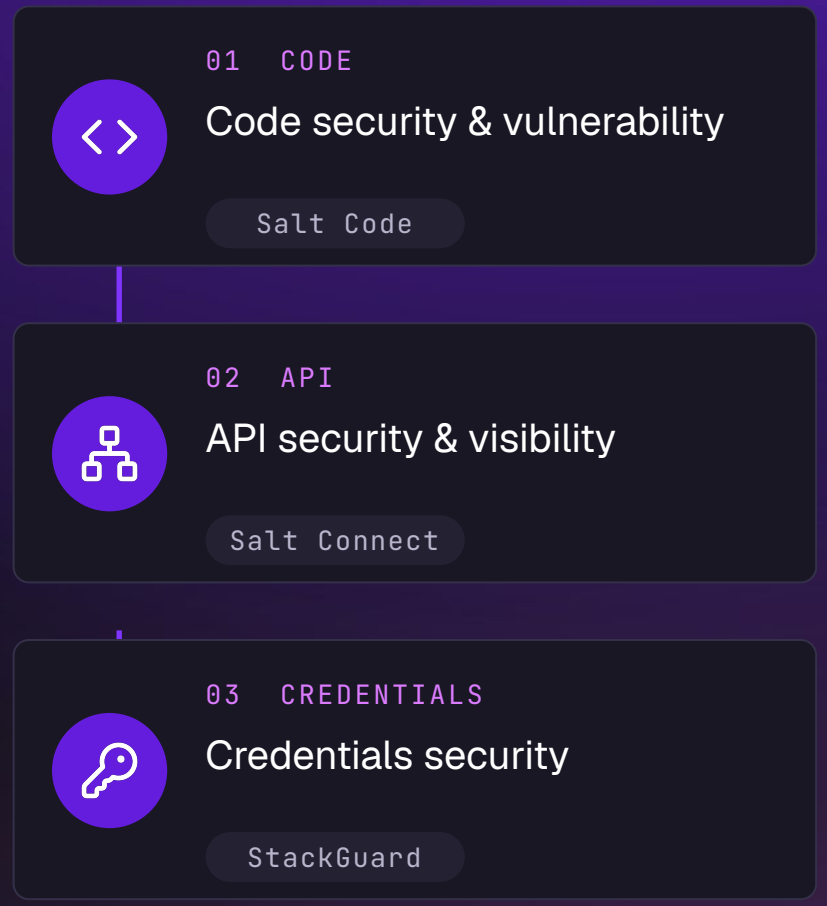
CONFERENCE SESSION · GOVT DIGITAL INFRASTRUCTURE

Securing your Code, API and Credentials

to improve the security posture for Govt digital infrastructure

Hemant Prasad

Founder & CEO, Crest Infosolutions Sdn Bhd



AGENDA

What we will cover

Three customer concerns, the capabilities that answer them, and how they work as one programme.

01

Code security & vulnerability

Govern AI-generated code at creation, in the pipeline and in runtime.

Salt Code

02

API security

See every API, MCP server and agent, and who owns it.

Salt Connect

03

Credentials security

Find, own and reduce machine credentials across the estate.

StackGuard

04

Better together

One lifecycle, one attack-chain view, one adoption roadmap.

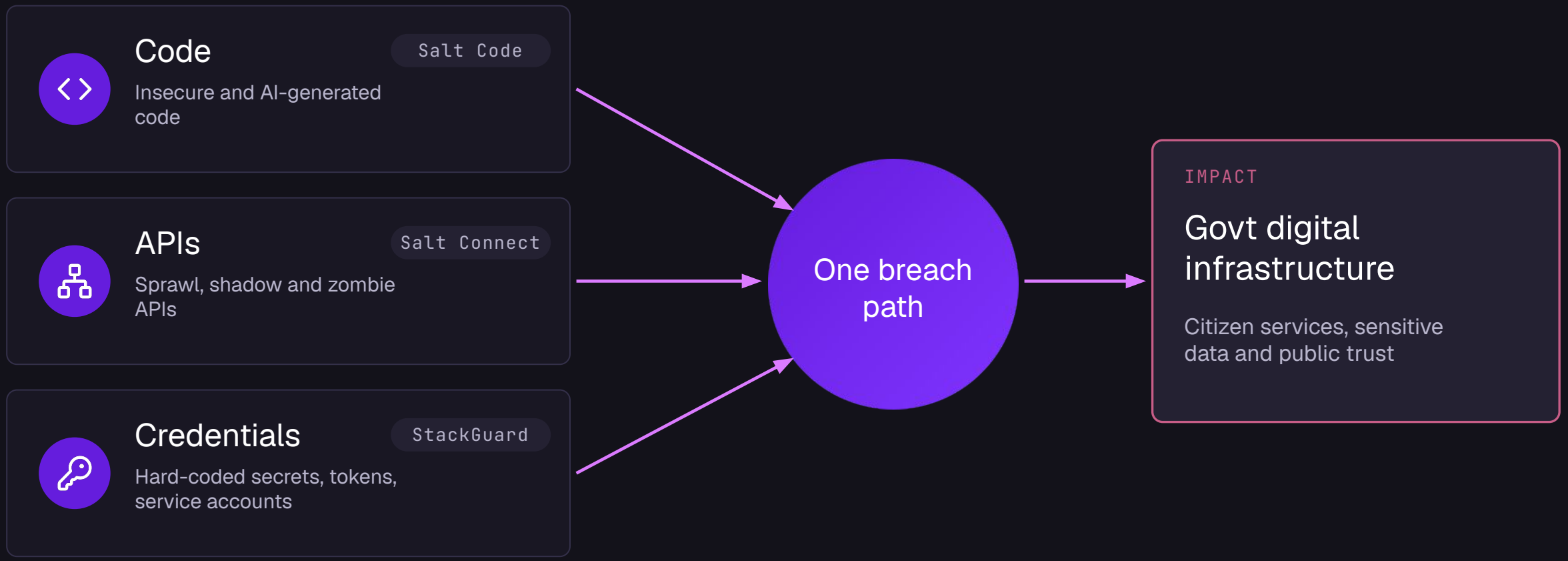
All three

THEN: ADOPTION ROADMAP · OUTCOMES · Q&A

WHY THEY CONVERGE

Three concerns, one breach path

Attackers chain weaknesses: a secret in code unlocks an API, and an unowned API exposes data.



WHY NOW

Why this is urgent

Three shifts have widened the attack surface faster than review and governance can follow.

AI writes the code

Assistants produce code faster than teams can review it, so insecure patterns ship unnoticed.

APIs, MCP servers and agents multiply

Teams ship new endpoints and agents continuously. Many have no owner, and dormant ones stay live.

Machine credentials hold standing access

Service accounts, keys, CI/CD tokens and OAuth grants rarely have an owner or a rotation plan.

THE THREE QUESTIONS THIS SESSION ANSWERS

Is the code we ship compliant with our policy?

Do we know every API we run, and who owns it?

Who owns each credential and what can it reach?

AI writes faster than anyone can review

Reviewing code after it is written cannot keep pace with AI-assisted development.

01

Insecure patterns ship unnoticed

Assistants repeat risky patterns at speed, such as weak authentication or missing validation, and reviews built for human pace miss them.

02

Security finds issues too late

Findings arrive after merge or release, when fixing is slowest and most disruptive for developers.

03

Policy enforcement is manual

Standards live in documents and checklists. Each team applies them differently and nobody can prove consistency.

CUSTOMER CONCERN

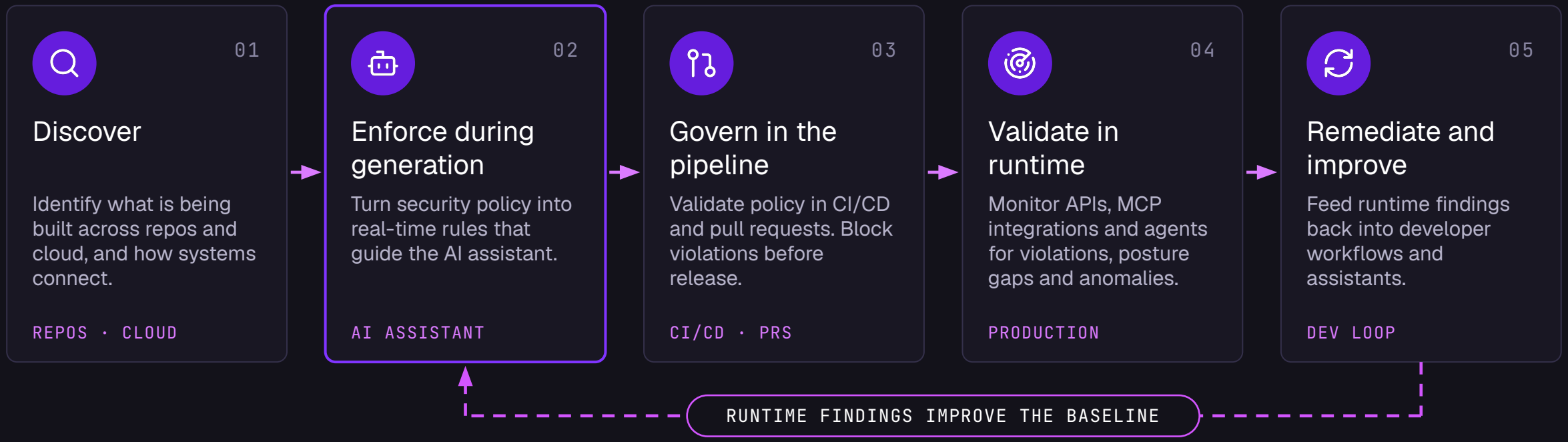
How do we keep AI-assisted code compliant with government security policy without slowing delivery?

ANSWER: SALT CODE

Policy enforcement from prompt to production

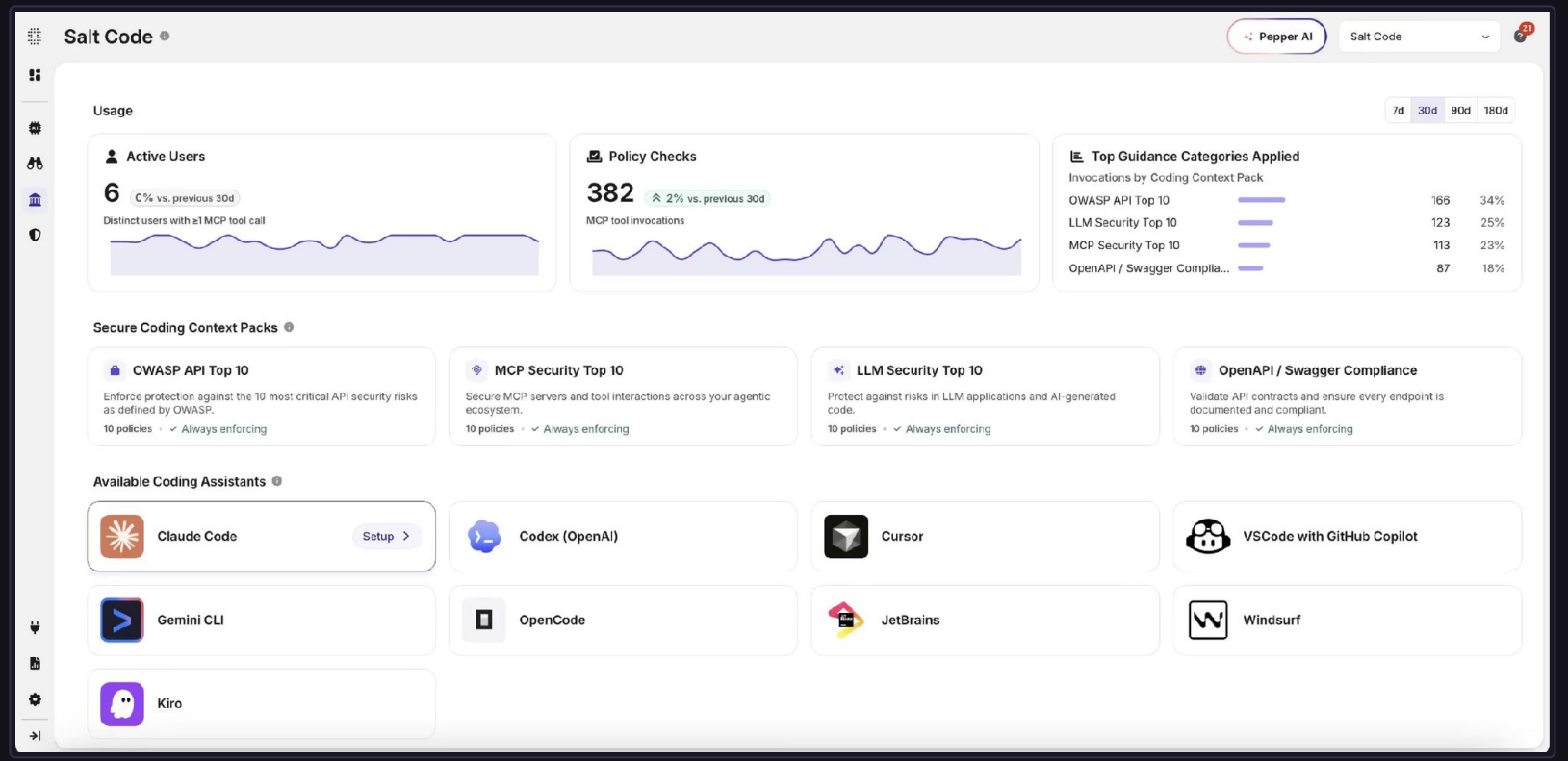
Salt Code turns security policy into rules the AI assistant follows, then checks them all the way to runtime.

UNIFIED GOVERNANCE One policy model for how APIs, MCP servers and agents are built, configured and validated



One console, every policy, every assistant

Central visibility into usage, secure coding context packs and the assistants you govern.



USAGE
Active users, MCP tool calls

CONTEXT PACKS
OWASP API, MCP, LLM Top 10

ASSISTANTS
Claude Code, Cursor, Copilot

140+

pre-built policies, grouped into secure coding context packs: OWASP API Top 10, MCP Security Top 10, LLM Security Top 10 and OpenAPI compliance.

USE CASES

- AI-generated APIs and MCP servers
- Shadow APIs stopped at the pull request
- Compliant code for every developer

Screenshot courtesy of Salt Security

You cannot govern what you cannot count

APIs, MCP servers and agents appear across clouds faster than teams can track them.

01

API sprawl, no owners

Assets spread across clouds, regions and gateways, and many have no named owner.

02

Dormant and zombie assets

APIs that are still live but no longer used or maintained stay exposed.

03

Basic questions, no answers

How many do we run, who owns them, and what do they expose? Few teams can say.

QUESTIONS MOST TEAMS CANNOT ANSWER QUICKLY

- How many APIs do we run?
- Who owns each one?
- Which are still used?
- What do they expose?

Salt Connect: inventory from the inside out

Read the cloud configuration instead of waiting for traffic to reveal what exists.

STEP 01

Connect read-only

Link AWS, Azure, GCP and GitHub. No agents, no traffic redirection.

STEP 02

Read configuration

Read metadata, logs and configuration: gateways, identities, scopes, reachable systems.

STEP 03

Build one inventory

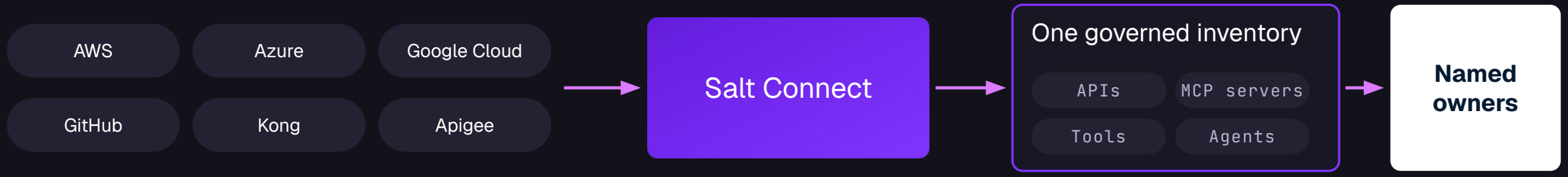
Consolidate APIs, MCP servers, tools and agents into one view.

STEP 04

Govern and assign

Flag zombie APIs, over-permissioned agents and misconfigured gateways. Route to owners.

READ-ONLY CONNECTORS · ZERO IMPACT ON TRAFFIC



What you get with Salt Connect

Six capabilities that turn a hidden API estate into a governed inventory.

Inside-out discovery

Find APIs, MCP servers and agents from cloud config.

Unified inventory

One view across clouds, regions and gateways.

Zombie API detection

Show assets that are live but unused or unowned.

Ownership and posture

See who owns each asset and how it is configured.

Agent & MCP governance

Spot over-permissioned agents and inactive MCP servers.

Fast, non-intrusive

Read-only connectors with zero impact on traffic.

SALT CONNECT · UNIFIED INVENTORY

All assets | APIs | MCP servers | Agents | Zombie

ASSET	TYPE	CLOUD	OWNER	STATUS
payments-api	API	AWS	Payments	Active
citizen-portal-gw	API	Azure	Digital Svcs	Active
orders-v1-legacy	API	AWS	Unassigned	Zombie
mcp-docs-search	MCP	GCP	Platform	Inactive
support-agent	Agent	Azure	CX team	Over-permitted
partner-api-v0	API	GCP	Unassigned	Deprecated

Illustrative view. Asset names are examples.

Salt cites **a full API inventory in under 30 minutes**, with no agents to deploy.

Five layers of API and agent risk

Salt covers each layer. Code and Connect are the two we focus on today.

EXPOSURE	Salt Surface	What attackers can reach from outside	
SOURCE CODE	Salt Code	Policy enforced as code is written	IN FOCUS
CONFIGURATION	Salt Connect	What exists and how it is governed	IN FOCUS
RUNTIME BEHAVIOR	Salt Collect	How APIs, MCP servers and agents behave	
RESPONSE	Salt Protect	Detects and blocks abuse in real time	

STACKGUARD

Credentials run through every layer

- Secrets in code and pipelines
- Tokens and keys behind each API
- Agent and workload permissions
- Ownership and audit evidence

Credentials are the quiet way in

Non-human credentials hold standing access across code, cloud, SaaS and AI tools, often without an owner.

Secrets in repositories and CI/CD

Pipeline tokens and keys sit in code, builds and deployment configuration.

Cloud workload identities

Roles assigned broadly while real use stays narrow.

SaaS OAuth tokens and API keys

Connected apps keep access long after the project ends.

Directory service accounts and certificates

Active Directory and Entra ID accounts with no reviewer.

AI agents and MCP access

Agents inherit downstream authority that nobody mapped.

WHAT GOES WRONG

- No owner to review or rotate
- Permissions broader than real use
- Stale credentials stay valid
- Evidence scattered across tools

Who owns each credential, and what can it reach?

ANSWER: STACKGUARD

StackGuard: six modules, one workflow

Purpose-built modules share one investigation workflow, so context is never lost.

01 / IDENTITY SECURITY

Non-human identity security

Discover service accounts, keys, tokens and machine identities with owner and lifecycle context.

02 / AI GOVERNANCE

AI agent security

Examine agent identities and permissions. Tie AI activity to SaaS scopes and owners.

03 / CLOUD SECURITY

Cloud & directory posture

Investigate cloud and directory findings with affected services and event records.

STACKGUARD PLATFORM · ONE INVESTIGATION WORKFLOW WITH IDENTITY CONTEXT INTACT

04 / INFRASTRUCTURE

Kubernetes & network insights

Follow signals to clusters, namespaces, pods, egress and pod-to-pod behavior.

05 / CREDENTIAL CONTROLS

Secrets & secretless access

Find exposed or stale credentials and cut reliance on static secrets.

06 / SECURITY OPERATIONS

Investigations, scans & reports

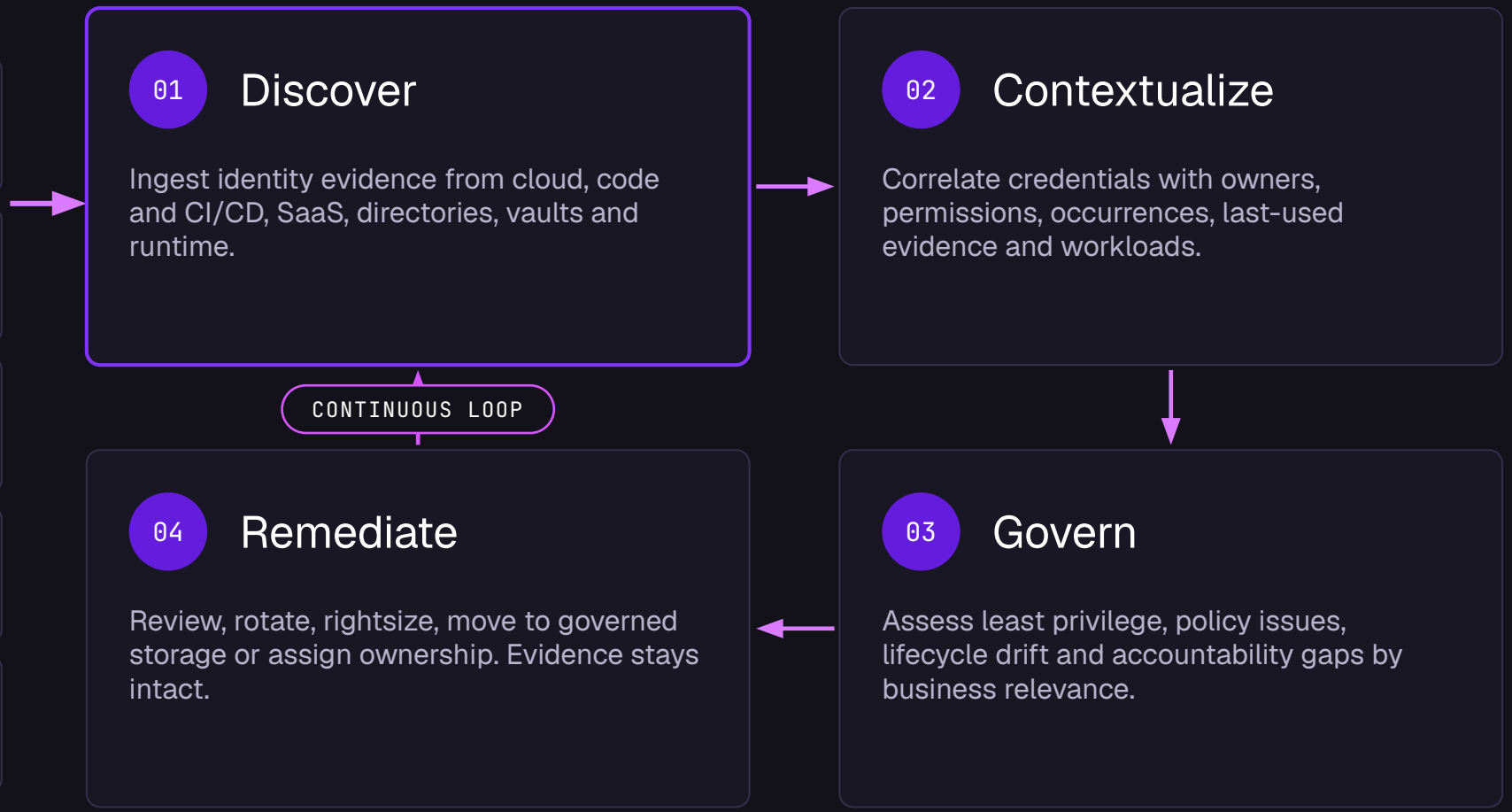
Move from finding to accountable action with history, scheduled scans and reports.

Discover, contextualize, govern, remediate

Four connected stages keep identity context intact from first signal to verified fix.

SOURCE FAMILIES

- **Cloud workloads** — AWS, Azure, Google Cloud
- **Code & CI/CD** — Repos, pipelines, build agents
- **SaaS applications** — OAuth grants, API tokens
- **Directories** — Active Directory, Entra ID
- **Secrets & response** — Vaults, SIEM, ticketing



See every identity and every agent

Ownership, last-used evidence and permission context for each machine identity.

STACKGUARD · NHI INVENTORY

All identities | Service accounts | Tokens | AI agents | Unowned

IDENTITY	TYPE	SOURCE	OWNER	LAST USED	RISK
ci-runner-token	CI/CD token	GitHub	Platform Eng	2 h ago	Critical
svc-payments	Service acct	AWS	Unassigned	41 d ago	High
oauth-crm-sync	OAuth grant	SaaS	Sales Ops	3 d ago	Medium
agent-support-bot	AI agent	Azure	CX team	1 h ago	High
vault-approle-etl	AppRole	Vault	Data Eng	12 d ago	Low
sa-legacy-reports	Service acct	Entra ID	Unassigned	187 d ago	High
sa-batch-etl	Service acct	GCP	Data Eng	6 d ago	Low

AI AGENT SECURITY

agent-support-bot

OWNER CX team

PERMISSION SET

crm.read	IN USE
tickets.write	IN USE
files.read.all	BROAD
mail.send	UNUSED
calendar.read	UNUSED

Recommended: remove 2 unused scopes

Illustrative StackGuard views

From signal to owner, evidence and response

Follow the evidence, not a trail of disconnected consoles.

1 Receive a signal
A finding, stale token or risky identity enters the queue.

2 Open identity context
Owner, last-used evidence, linked systems and permissions.

3 Assess blast radius
What it can reach, and whether the risk is isolated or broad.

4 Choose the action
Rotate, revoke, rightsize, move to Vault or assign an owner.

5 Track and verify
Record who reviewed it, what changed and the risk reduced.

STACKGUARD · FINDING DETAIL

FINDING / NHI-2048

Over-privileged CI runner token

CRITICAL · 9.4

Overview | Posture | Occurrences | Lifecycle | Audit history

CI token → IAM → Cloud → K8s → Response

01 · Discovery Credential connected to service identity and owner.

02 · Assessment Least-privilege review finds unused actions.

OWNER
Platform Engineering

UNUSED PERMISSIONS
21 actions

RELATED EVENTS
8 occurrences

RECOMMENDED RESPONSE
Rightsize + rotate

Review & rotate

Beyond vaults: fewer static secrets

StackGuard complements your secrets managers and moves access toward governed, secretless patterns.

Find exposed and stale credentials

Across repos, pipelines, cloud and SaaS.

Evaluate the right remediation

Rotate, retire, rightsize, move to governed storage or go secretless.

Keep evidence tied to action

Every change is recorded with who, what and why.

WORKS WITH YOUR SECRETS ECOSYSTEM

- HashiCorp Vault
- Azure Key Vault
- AWS Secrets Manager
- Google Cloud Secret Manager

STACKGUARD · SECRETS & REMEDIATION

REMEDIATION PROPOSALS

CREDENTIAL	FOUND IN	STATE	PROPOSAL
ci-deploy-token	Pipeline variable	Exposed	**Rotate**
svc-reporting-key	Cloud workload	Stale 94 d	**Retire**
etl-db-password	Config file	Static	**Move to Vault**
partner-api-key	SaaS integration	Over-scoped	**Rightsize**
batch-job-secret	Kubernetes	Static	**Go secretless**

Illustrative view

One lifecycle, three layers of control

Where each capability acts between the first prompt and production.

	CREATE AI assistant · IDE	COMMIT & BUILD Repo · PR · CI/CD	DEPLOY Cloud · gateways	RUN APIs · MCP · agents
Salt Code Code security	Enforce policy during generation	Pipeline gates block violations before release		Validate APIs, MCP and agents in runtime
Salt Connect API security			Read-only discovery of APIs, MCP servers, agents	One governed inventory, owners assigned
StackGuard Credentials security	Map agent permissions and scopes	Find exposed secrets in repos and pipelines	Discover credentials, owners, permissions	Rightsize, rotate or go secretless

SHARED OUTCOME: ONE POLICY, ONE INVENTORY, ONE OWNER AND EVIDENCE TRAIL PER ASSET

Breaking the chain at every link

Illustrative scenario: one leaked token, followed from AI-generated code to data exposure.

WHAT HAPPENS WITHOUT CONTROLS

1. **AI writes an endpoint**
   Generated code hard-codes a token and skips an auth check.
2. **The code ships**
   The token sits in the repo and pipeline. Review misses it.
3. **API goes live, unowned**
   No named owner, and it is never retired.
4. **Attacker abuses it**
   The leaked token reaches the API and its excess permissions.
5. **Impact**
   Citizen data exposed. Response slowed by missing ownership.

WHERE EACH LAYER BREAKS THE CHAIN

SALT CODE
Policy guides the assistant, so no token is hard-coded and auth is built in.

STACKGUARD
Exposed credential found. Owner and blast radius mapped. Rotate or go secretless.

SALT CONNECT
Unowned or zombie API found from cloud config. Assigned or retired.

STACKGUARD
Excess permissions rightsized, so a stolen token reaches far less.

Chain broken
Data stays protected, and every action is logged with evidence.

Customer concern to capability map

What to ask, what answers it, and what the customer gets.

CUSTOMER CONCERN	CAPABILITY	PRODUCT	WHAT THE CUSTOMER GETS
Is AI-generated code compliant with our policies?	Policy enforced inside AI assistants, plus PR and CI/CD gates	SALT CODE	Compliant by default. Violations blocked before release.
Do we know every API, MCP server and agent we run?	Inside-out cloud discovery into one inventory	SALT CONNECT	One inventory across clouds, regions and gateways.
Which APIs are dead but still live?	Zombie and deprecated API detection with owner routing	SALT CONNECT	Dormant assets retired or assigned an owner.
Who owns each credential, and what can it reach?	NHI discovery, ownership and permission analysis	STACKGUARD	Owner, last-used evidence and blast radius per identity.
Are secrets exposed in repos and pipelines?	Exposure detection, remediation proposals, secretless access	STACKGUARD	Rotate, rightsize or move to governed storage.
Are AI agents over-permissioned?	Agent permission review and MCP governance	SALT + STACKGUARD	Agent access mapped to owners and scopes.

Governance you can evidence

Align controls to recognised standards and keep an audit trail for every action.

SALT CODE

Standards-aligned policy

- OWASP API Security Top 10 policies
- MCP Security Top 10 policies
- Custom rules for local policy
- EU AI Act referenced in Salt governance context

SALT CONNECT

Inventory and posture evidence

- Named owner for every asset
- Configuration and gateway posture
- Deprecated and zombie asset records

STACKGUARD

Identity audit trail

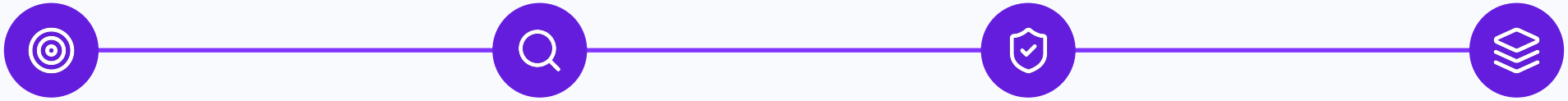
- Audit history on every finding
- Who reviewed it and what changed
- Reports, activity logs and scheduled scans

Mapping to your national cyber and data-protection framework is confirmed with you during scoping.

ADOPTION ROADMAP

From a focused pilot to enterprise coverage

An indicative 90-day path. We confirm scope and acceptance criteria with you first.



SCOPE

Weeks 0-2

- Walk through your estate
- Pick priority use cases
- Agree acceptance criteria

DISCOVER

Days 1-30

- Connect cloud and GitHub read-only
- Baseline APIs and credentials
- Name owners for top risks

GOVERN

Days 31-60

- Enable policy in a pilot team
- Add PR and CI/CD gates
- Start rotate and rightsize

SCALE

Days 61-90

- Extend to all teams
- Validate in runtime
- Report outcomes monthly

DEPLOY ON YOUR TERMS: ON-PREMISES, PRIVATE CLOUD, OR MICROSOFT MARKETPLACE FOR AZURE (STACKGUARD)

Outcomes and how we measure them

Baseline in the first 30 days, then track every month.

Fewer insecure patterns reach review

KPI

Policy violations blocked at generation and PR

A complete API inventory

KPI

Share of APIs with a named owner

Zombie APIs retired

KPI

Dormant APIs found vs retired

Less credential exposure

KPI

Exposed and stale secrets still open

Least privilege in practice

KPI

Unused permissions removed

Faster, evidenced response

KPI

Time from finding to owner and action

Qualitative outcomes today. Targets are set with you once the baseline is measured.

Q&A

Questions?

Conversation starters

- AI-generated code governance
- API inventory and zombie APIs
- Credential ownership and secretless access

Hemant Prasad · Founder & CEO, Crest Infosolutions Sdn Bhd
info@crestsolution.com · crestsolution.com

CREST

— Ideate — Innovate — Implement —

Thank you

Secure the code, the APIs and the credentials behind digital government.

Salt Code

CODE SECURITY

Salt Connect

API SECURITY

StackGuard

CREDENTIALS SECURITY

Hemant Prasad

Founder & CEO, Crest Infosolutions Sdn Bhd · info@crestsolution.com



CREST
Ideate · Innovate · Implement

PUBLIC SECTOR TECH DAY 2026

12 NOV 2026 | GARDEN BALLROOM, LEVEL 1
MARIOTT HOTEL, PUTRAJAYA

Spill the Tea!

CREST
Ideate · Innovate · Implement

PUBLIC SECTOR TECH DAY 2026

12 NOV 2026 | GARDEN BALLROOM, LEVEL 1
MARIOTT HOTEL, PUTRAJAYA

We'd Love to Hear From You!
Send Us Your Feedback

