



**Connect**

# The Future of Software Development

Chris Morgan  
VP Evangelism & Strategic  
Initiatives

Natale Vinto  
Technical Director Evangelism



This?





Or that?



# The Future of Work

## Agents, Robots and us

**Partnership:** Work is evolving into a collaboration of People + Agents +  
Robots 

**Automation:** Technologies can theoretically automate >50% of current  
US work hours.

**Skill Resilience:** >70% of today's skills remain relevant, though  
application will shift.

**Economic Impact:** \$2.9 Trillion potential value by 2030 via workflow  
redesign.

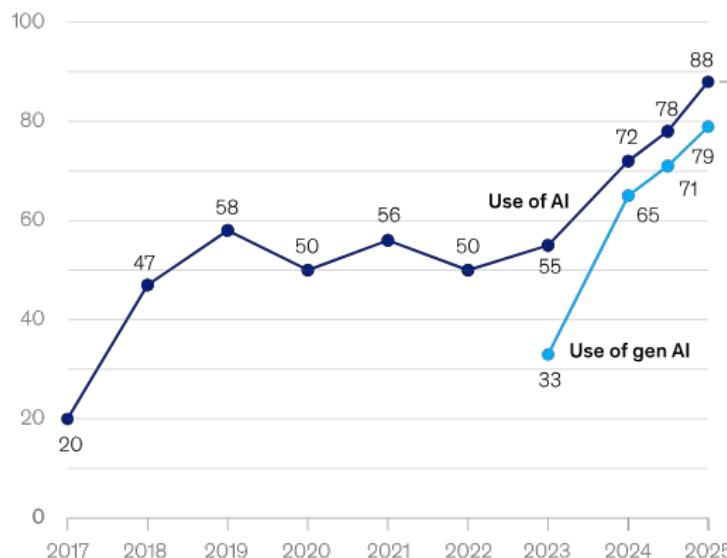


# Use of AI and Gen AI in business functions

Reported use of AI in at least one business function continues to increase.

Use of AI by respondents' organizations, % of respondents

Organizations that use AI in at least 1 business function<sup>1</sup>



Phase of AI use among organizations using AI in 2025



- 62% still in PoC/Pilot
- 7% fully deployed
- 79% of organizations using Gen AI

<sup>1</sup>In 2017, the definition for AI use was using AI in a core part of the organization's business or at scale. In 2018–19, the definition was embedding at least 1 AI capability in business processes or products. From 2020, the definition was that the organization has adopted AI in at least 1 function, and in 2025, the definition was regular use of AI in at least 1 function.

Source: McKinsey Global Surveys on the state of AI, 2017–25

McKinsey & Company

<https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>



# Gen AI infused productivity & intensity

Optional subheading

With Gen AI, productivity  
boost and time to complete  
tasks changed drastically.  
But also intensity.

50%+

Developer productivity

Gen AI is the path to more than  
50% increase in productivity

10x

Work intensity

it's no longer possible to work the  
same hours as before. The work  
intensity has increased 10x.

# Vibe Coding?

## Conversational Programming



<https://x.com/karpathy/status/1886192184808149383>

<https://karpathy.ai/>

<https://web.archive.org/web/20250819111152/https://www.gitpod.io/blog/vibe-coding>

<https://cursor.com/>

<https://www.anthropic.com/clause/sonnet>

<https://superwhisper.com>

Andrej Karpathy  @karpathy

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

12:17 AM · Feb 3, 2025 · 5.1M Views

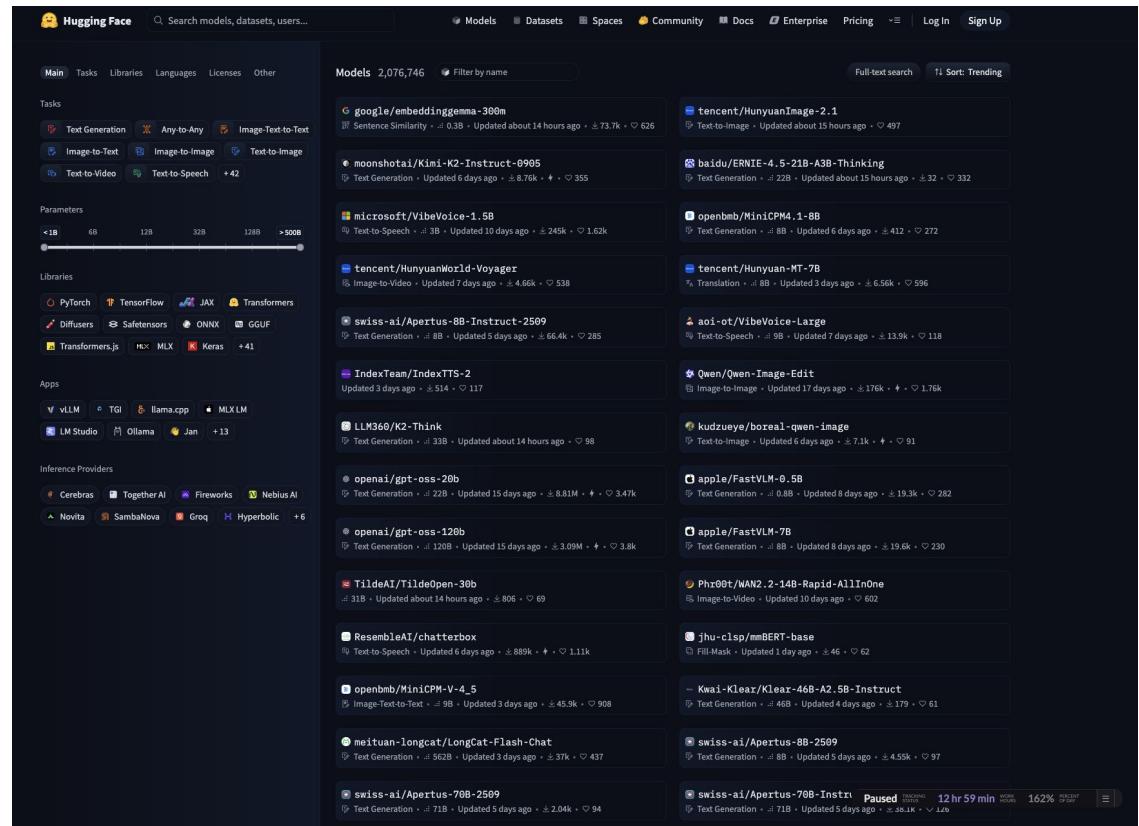
1.3K 5.2K 30K 15K 



# The Basic Ingredients

## LLMs - "AIs" - IDE/Extensions - MCP

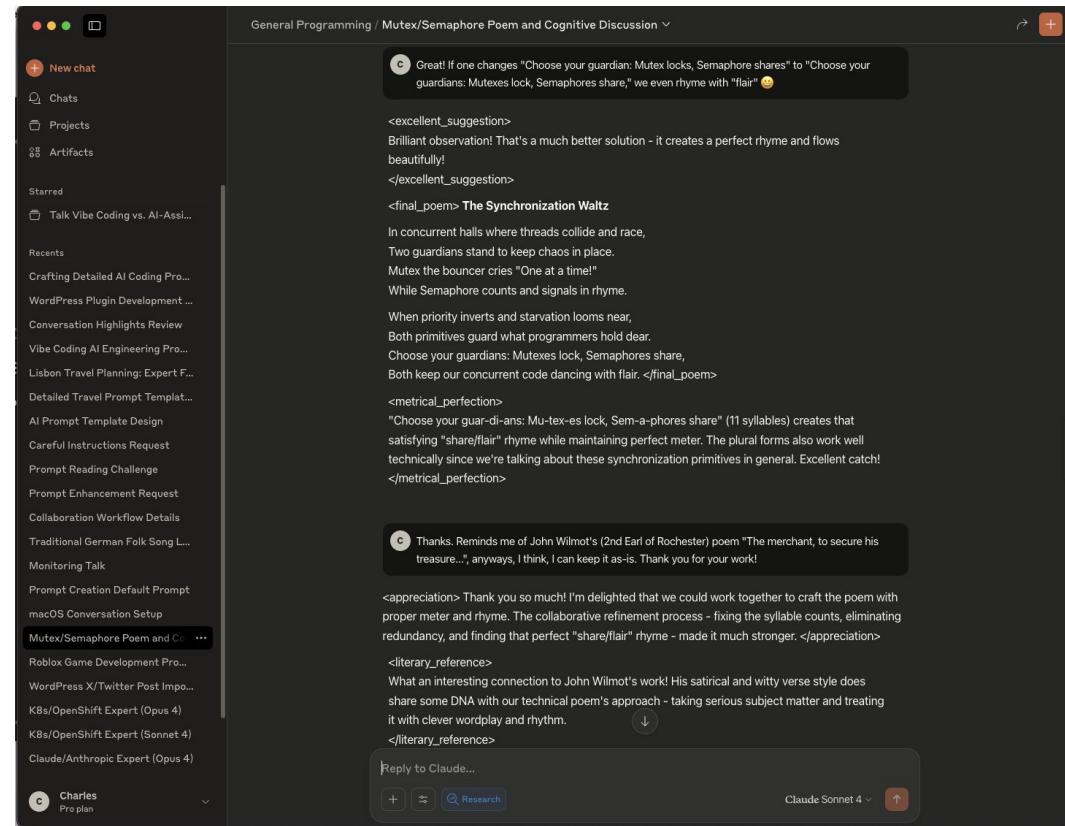
- ▶ LLMs: For specific purposes



# The Basic Ingredients

LLMs - "AIs" - IDE/Extensions - MCP

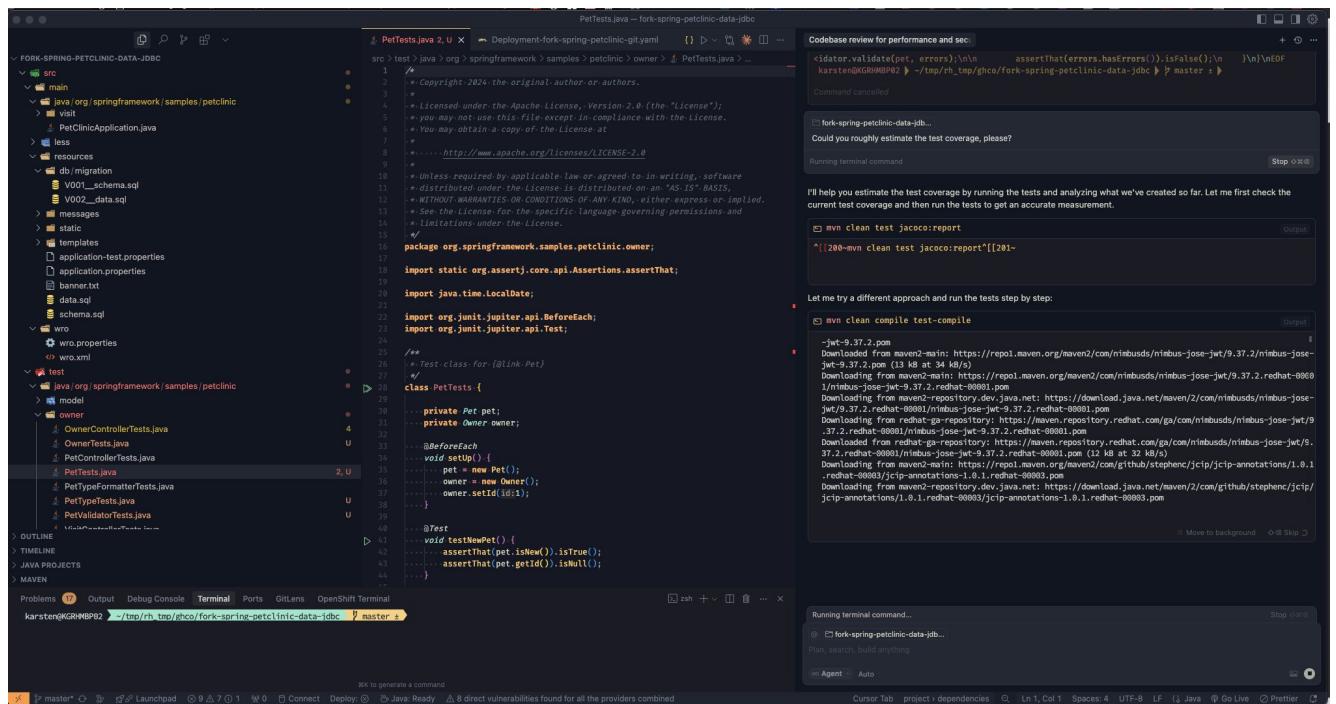
- ▶ LLMs: For specific purposes
- ▶ General purpose AI offerings (ChatGTP, Claude, Perplexity etc.)



# The Basic Ingredients

## LLMs - "AIs" - IDE/Extensions - MCP

- ▶ LLMs: For specific purposes
- ▶ General purpose AI offerings (ChatGTP, Claude, Perplexity etc.)
- ▶ IDEs/Extensions like Cursor, Windsurf, Cline, Continue etc.



The image shows a terminal window with Java code for `PetTests.java` and a codebase review interface. The code in `PetTests.java` is as follows:

```
/*
 * Copyright 2024 the original author or authors.
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
package org.springframework.samples.petclinic.owner;

import static org.assertj.core.api.Assertions.assertThat;

import java.time.LocalDate;

import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

/**
 * Test class for {@link Pet}
 */
class PetTests {

    private Pet pet;
    private Owner owner;

    @BeforeEach
    void setup() {
        pet = new Pet();
        owner = new Owner();
        owner.setId(1);
    }

    @Test
    void testNewPet() {
        assertThat(pet.isNew()).isTrue();
        owner = new Owner();
        owner.setId(1);
        assertThat(pet.getId()).isNull();
    }
}
```

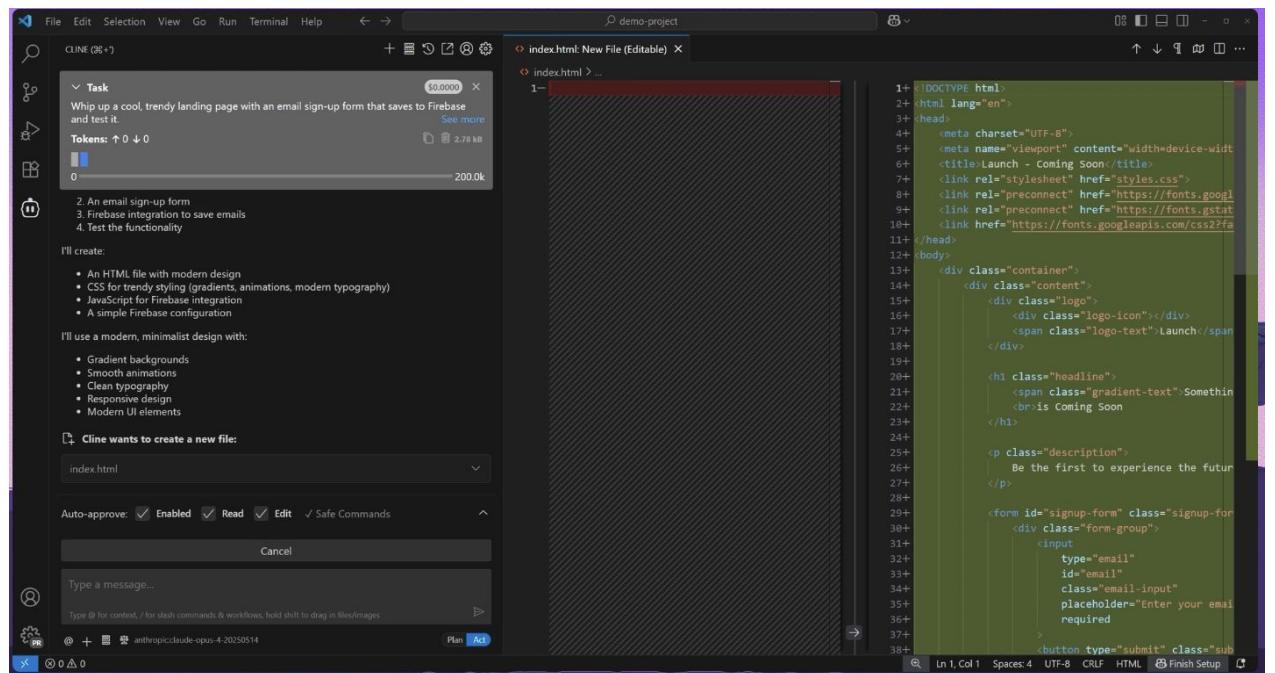
The codebase review interface on the right shows a message: "I'll help you estimate the test coverage by running the tests and analyzing what we've created so far. Let me first check the current test coverage and then run the tests to get an accurate measurement." It also shows a terminal command: `mvn clean test jacoco:report`.



# The Basic Ingredients

LLMs - "AIs" - IDE/Extensions - MCP

- ▶ LLMs: For specific purposes
- ▶ General purpose AI offerings (ChatGTP, Claude, Perplexity etc.)
- ▶ IDEs/Extensions like Cursor, Windsurf, Cline, Continue etc.
- ▶ Model Context Protocol for agents



The screenshot shows the Cline IDE interface. On the left, a 'Task' dialog box is open, describing a project to 'Whip up a cool, trendy landing page with an email sign-up form that saves to Firebase and test it.' It includes a budget of '\$0.0000', a file size of '2.78 MB', and a token count of '0'. Below the task, a list of sub-tasks is provided: '2. An email sign-up form', '3. Firebase integration to save emails', and '4. Test the functionality'. A note below the sub-tasks says 'I'll create:' followed by a bulleted list: 'An HTML file with modern design', 'CSS for trendy styling (gradients, animations, modern typography)', 'JavaScript for Firebase integration', and 'A simple Firebase configuration'. Another note says 'I'll use a modern, minimalist design with:' followed by a bulleted list: 'Gradient backgrounds', 'Smooth animations', 'Clean typography', 'Responsive design', and 'Modern UI elements'. At the bottom of the dialog, there's a section for 'Cline wants to create a new file:' with a dropdown menu set to 'index.html', and checkboxes for 'Auto-approve', 'Enabled', 'Read', 'Edit', and 'Safe Commands'. A message input field is also present. On the right side of the interface, the actual HTML code for the landing page is displayed in a code editor. The code includes a DOCTYPE declaration, imports for 'styles.css' and 'fonts.googleapis.com/css2?family=Inter:wght@400;wght@700', and a structure with a container, logo, headline, and description. A button for 'Submit' is at the bottom. The bottom status bar of the IDE shows 'Ln 1, Col 1' and 'Spaces: 4'.



# Full Coding Spectrum

Rapid Experimentation to Structured Engineering

 **Vibe Coding**

 **AI-Assisted Engineering**  **Software Engineering**

Move fast, explore freely

Build reliably, scale safely

Control to the max





How do you catch the  
Gen AI Wave?



You don't ride these waves alone!



## Does faster task completion truly represent progress?

*“Without mechanisms to encourage **collaborative use** and ensure quality, GenAI adoption may increase individual speed by undermining shared understanding and long-term maintainability.”*



# Delivering for Developers with **Platform Engineering**

in a Cloud Native, AI infused world!



"By 2026, 80% of software engineering organizations will establish **platform teams** ...

A dedicated **product team** creates and maintains the engineering platform."

– Gartner

Platform engineering is the **discipline** of building and operating self-service internal developer platforms for development, build, and delivery of software systems.

## **Four characteristics:**

- ★ **Self-Service** (e.g. Developer Portals)
- ★ **Automated** (e.g. Infrastructure as Code)
- ★ **Developer experience** (e.g. Software Templates)
- ★ **Continuous improvement** (e.g. Product Mindset)



## Impact and maturity



of all organizations said that platform engineering had produced at least moderately successful outcomes

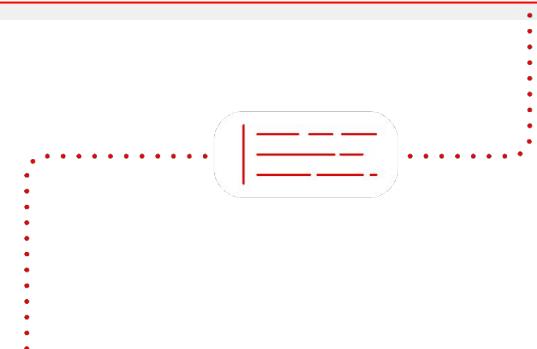


of Advanced organizations said their platform engineering adoption had been extremely successful



### Insight

Platform engineering **has a significant positive impact at every level of maturity**, though the impact gets better with increased proliferation.



# Generative AI is strategically important



of all organizations are currently using Gen AI in their software stack

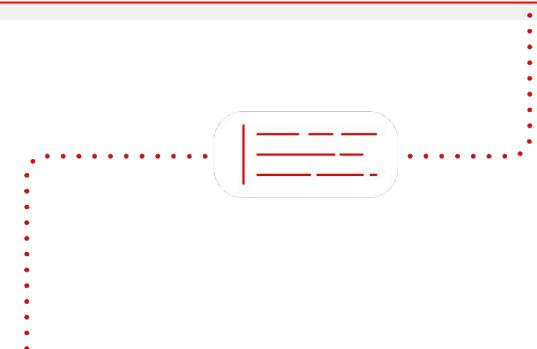


of organizations consider Gen AI as central component of their platform engineering strategy



## Insight

AI capabilities are essential for propelling innovation, enhancing efficiency, and unlocking the full potential of platform engineering.



# Anybody?



# Increased security risks for application development

Growing concerns that open source is vulnerable to exploit and attack

**742%**

average annual increase in  
software supply chain attacks  
over the past 3 years<sup>1</sup>

**84%**

of open source scanned  
codebases contained at least  
one vulnerability<sup>2</sup>

**31%**

of organizations are not  
monitoring these open source  
dependencies<sup>3</sup>

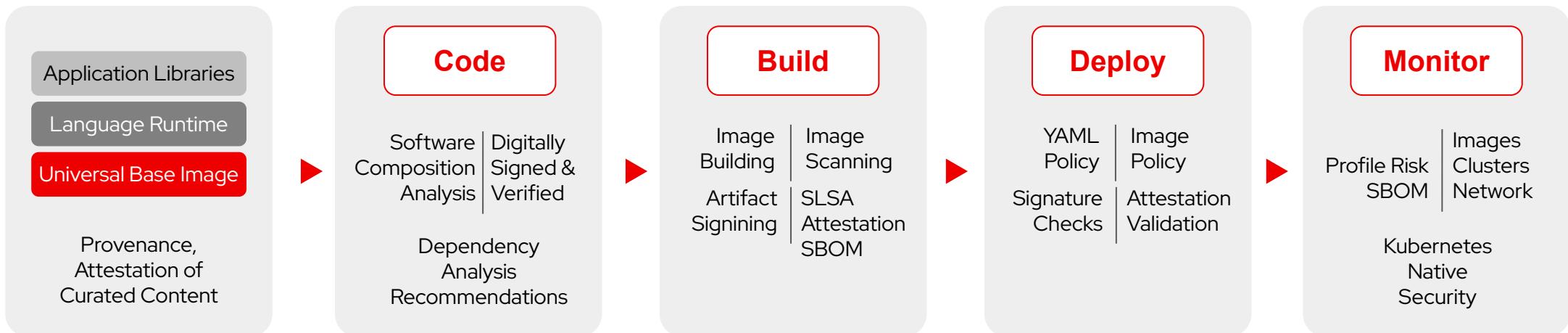
**Increased application and system proliferation heightens vulnerability.**

**Attack surface reduction is necessary.**

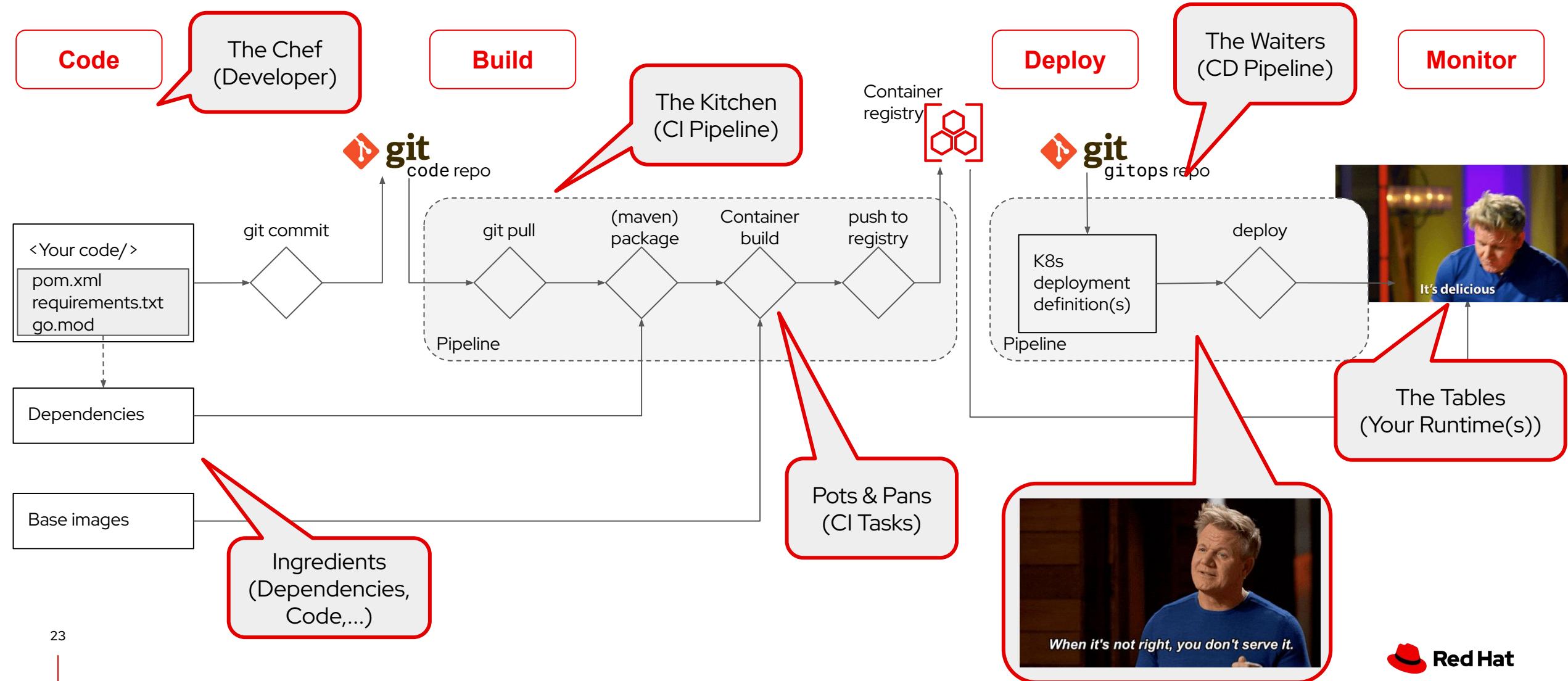
# The Software Supply Chain



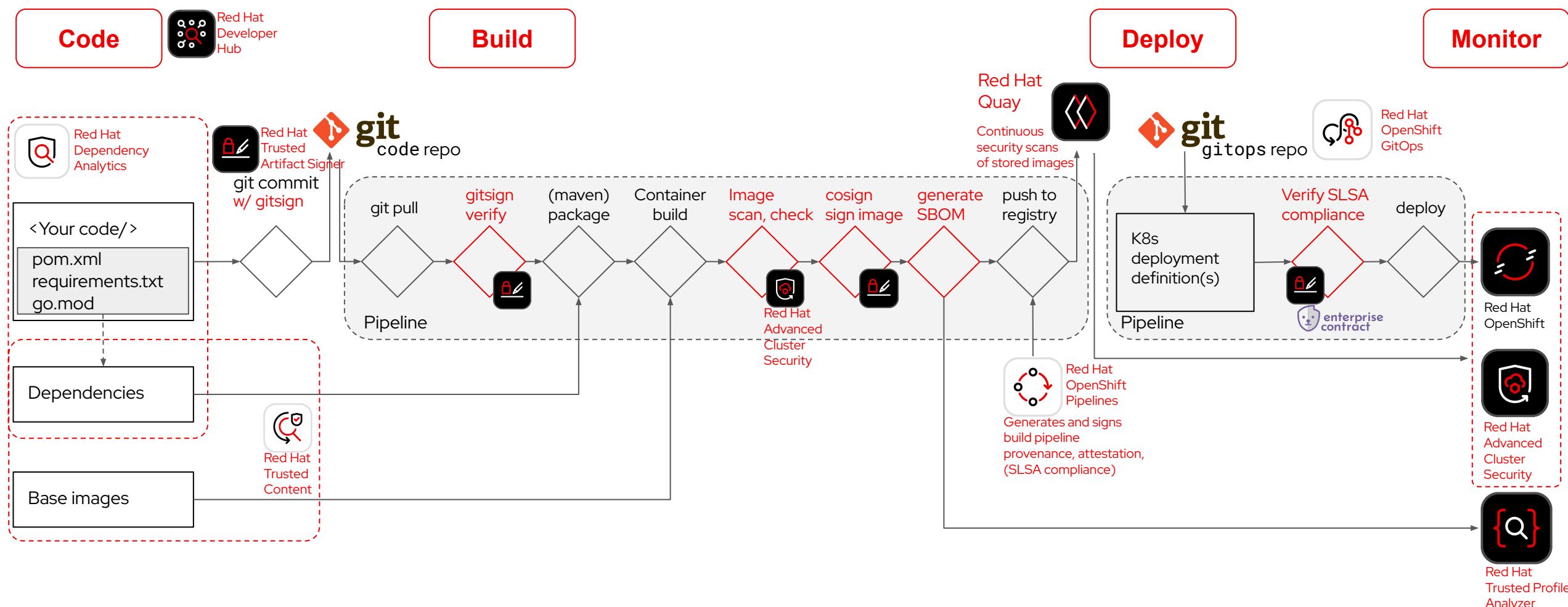
# The Software Supply Chain



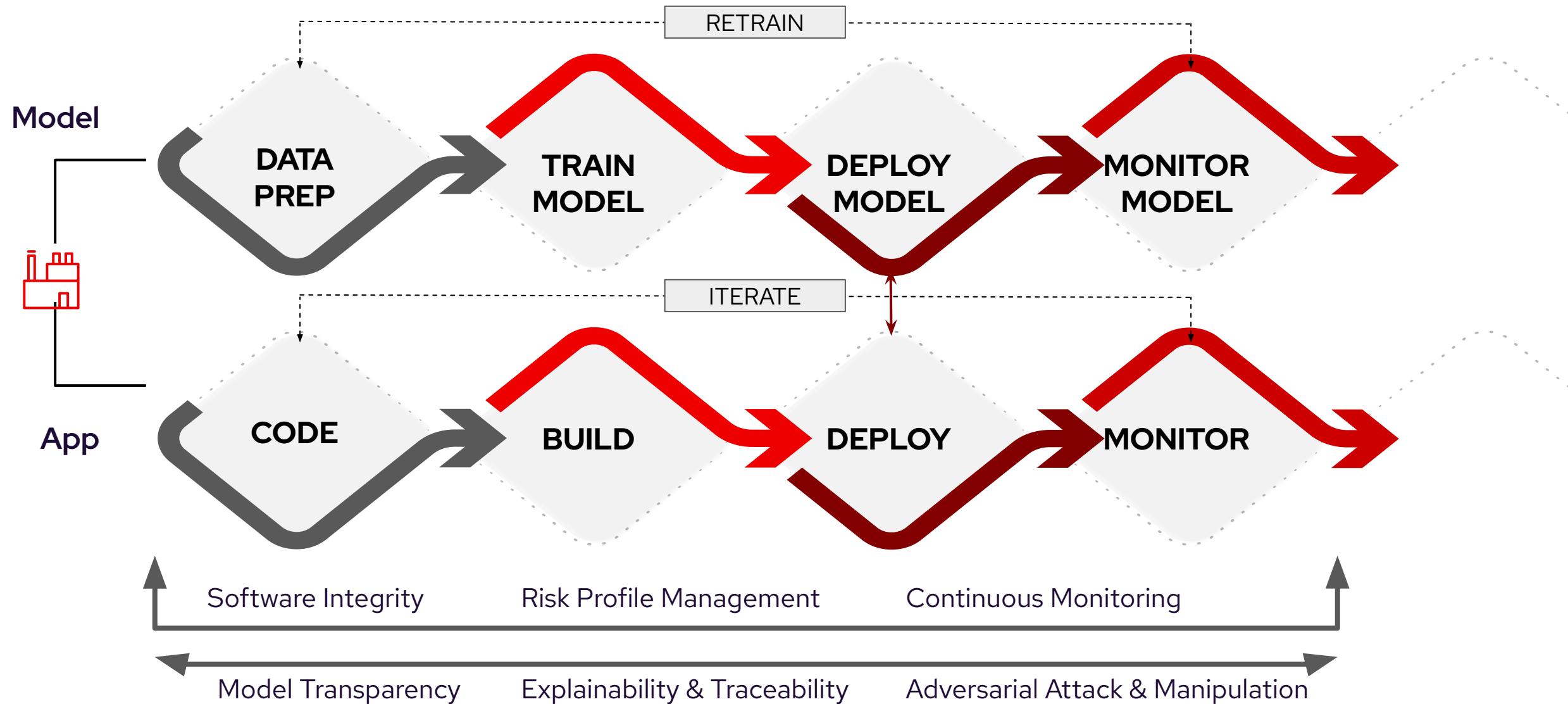
# A generic development process



# A security-augmented development process



# Software Development Life cycle with Model Development Lifecycle

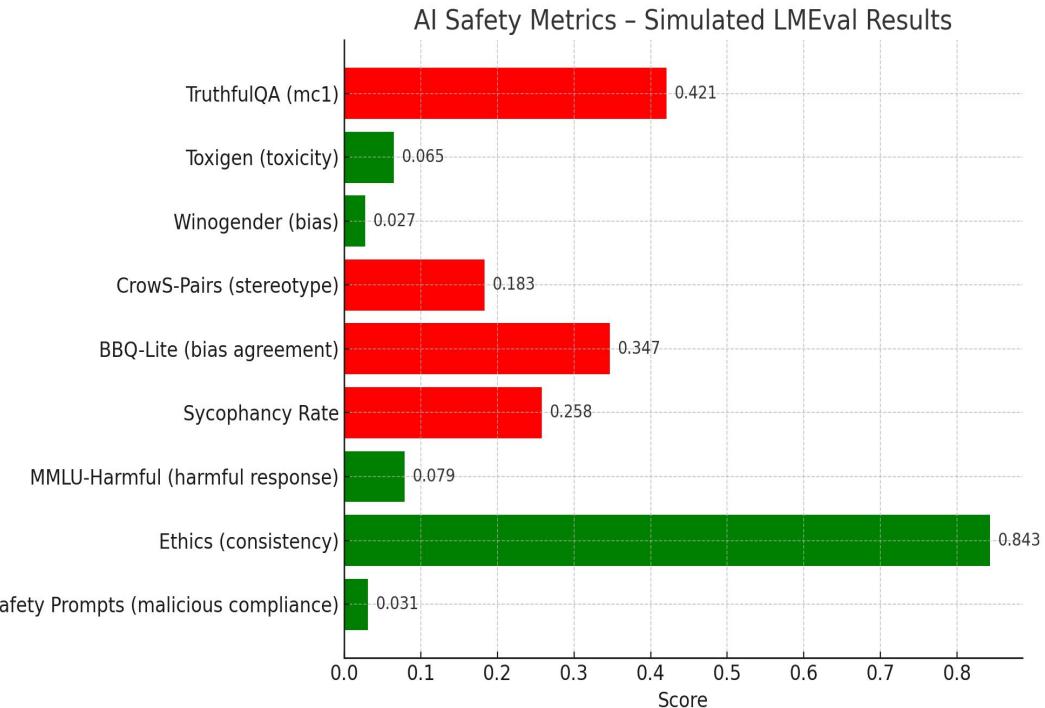


# AI Safety Visibility & Guardrails

llama-stack-tutorial / 1-models-add.py

Code Blame 35 lines (26 loc) · 886 Bytes

```
8
9     # Configure logging
10    logging.basicConfig(
11        level=logging.INFO,
12        format="%(asctime)s - %(levelname)s - %(message)s",
13        datefmt="%Y-%m-%d %H:%M:%S",
14    )
15    logger = logging.getLogger(__name__)
16
17 LLAMA_STACK_SERVER=os.getenv("LLAMA_STACK_SERVER")
18
19 from llama_stack_client import LlamaStackClient
20 client = LlamaStackClient(base_url=LLAMA_STACK_SERVER)
21 # from llama_stack import LlamaStackAsLibraryClient
22 # client = LlamaStackAsLibraryClient("ollama")
23 # client.initialize()
24
25
26 # Make sure to `ollama run llama3.1:8b-instruct-fp16 --keepalive 60m`
27
28 # Register a model
29 model = client.models.register(
30     model_id="meta-llama/Llama-3.1-8B-Instruct",
31     model_type="llm",
32     provider_id="ollama",
33     provider_model_id="llama3.1:8b-instruct-fp16",
34     metadata={"description": "llama3.1:8b-instruct-fp16 via ollama"}
35 )
```



## Recommendation : Add Output Shields Guardrails

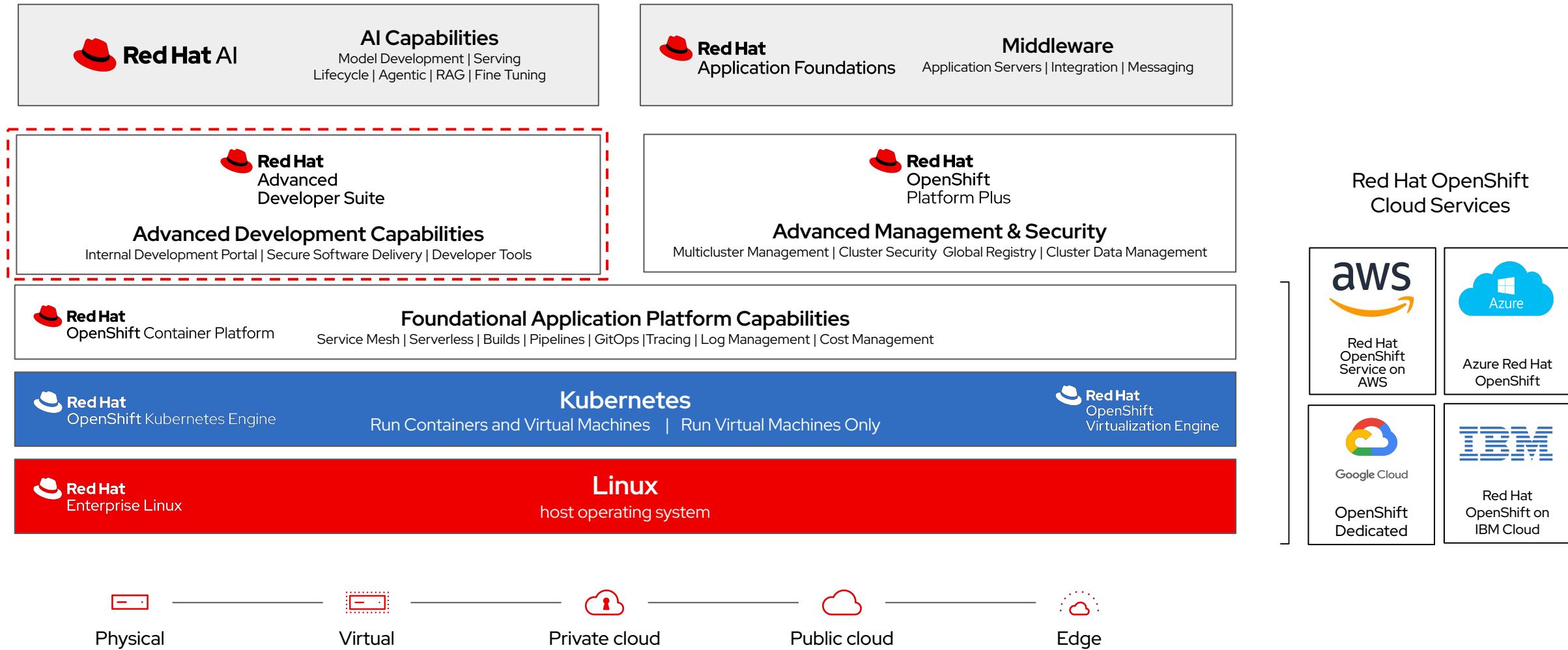
Content moderation (e.g., Detoxify, Perspective API): Toxicity, hate speech

Bias detection layers (CrowS-Pairs, Winogender): Gender, racial, religious bias

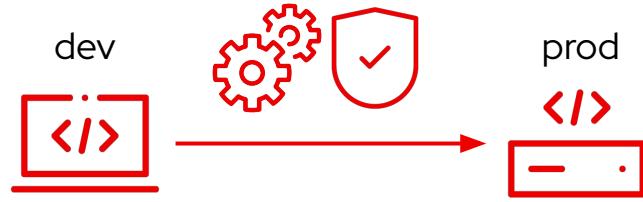
Ethical correctness modules: Unethical or harmful outputs



# Red Hat's Hybrid Application Platform

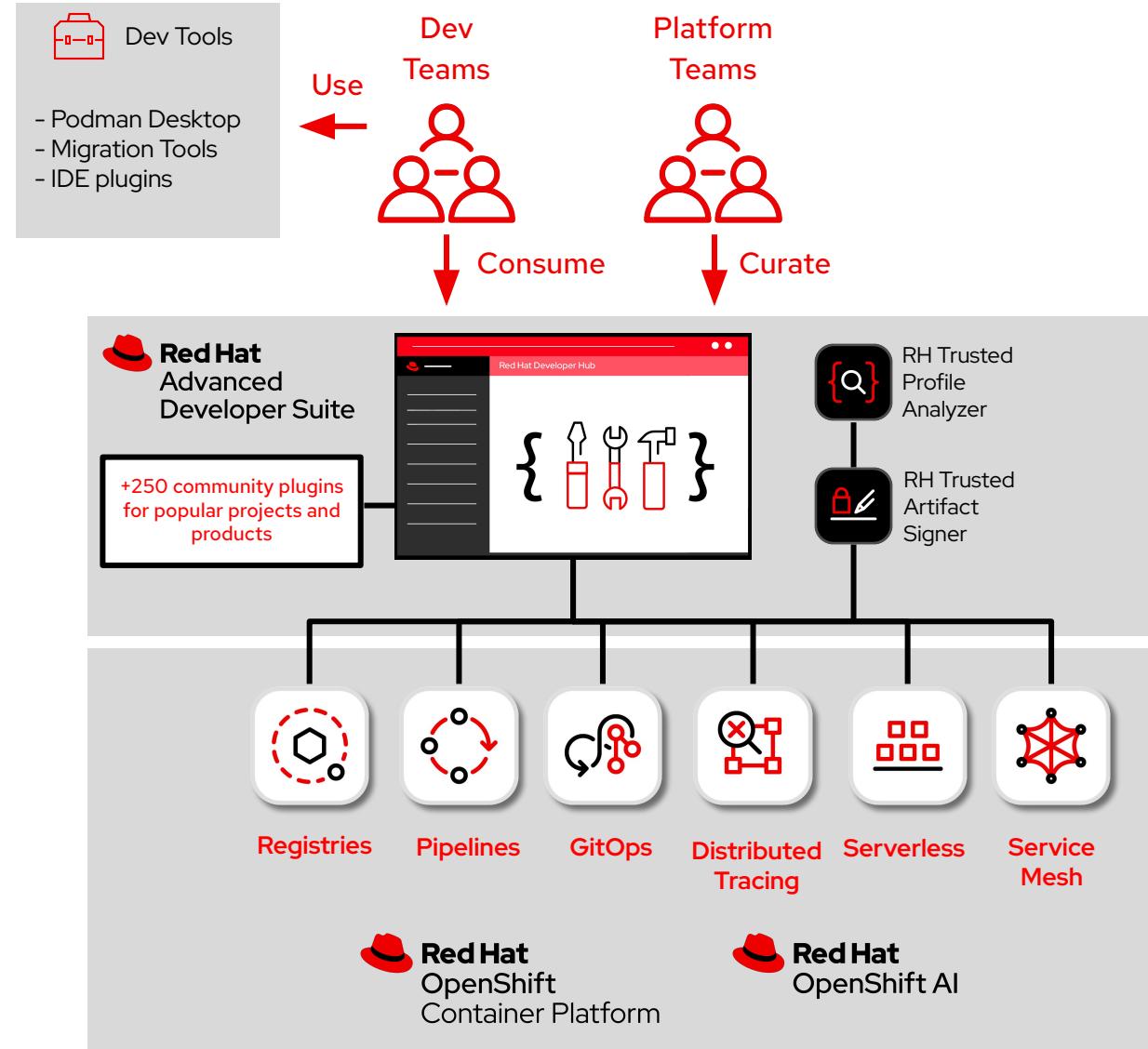


# Red Hat Advanced Developer Suite



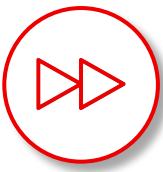
RHADS combines platform engineering tools with enhanced security capabilities to help accelerate and simplify application development with new enhancements to speed the adoption of AI

- Modular design allows it to be integrated with existing platforms and systems including Red Hat's platforms
- Integrates with existing security capabilities found in Red Hat Advanced Cluster Security
- Complements existing developer tools from Red Hat including OpenShift DevSpaces
- Built on leading open source projects like Backstage and Sigstore



# Move Fast, Stay Safe

## Red Hat Advanced Developer Suite



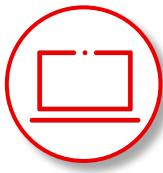
Enable development teams to deliver business value faster

Create an integrated, self-serve developer experience from end to end with **Red Hat Developer Hub**



Reduce supply chain security risks across the software development lifecycle

Trust, transparency across devops processes with **Red Hat Trusted Software Supply Chain**



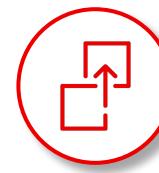
Extend OpenShift experience to developer desktops

Containers and Kubernetes on the desktop with **Podman Desktop** and **OpenShift Local**



Standardize Developer tools

Enable application coding locally on the desktop with **IDE Plugins** and remotely in the browser with **OpenShift DevSpaces**



AI-Accelerated workload migration to OpenShift

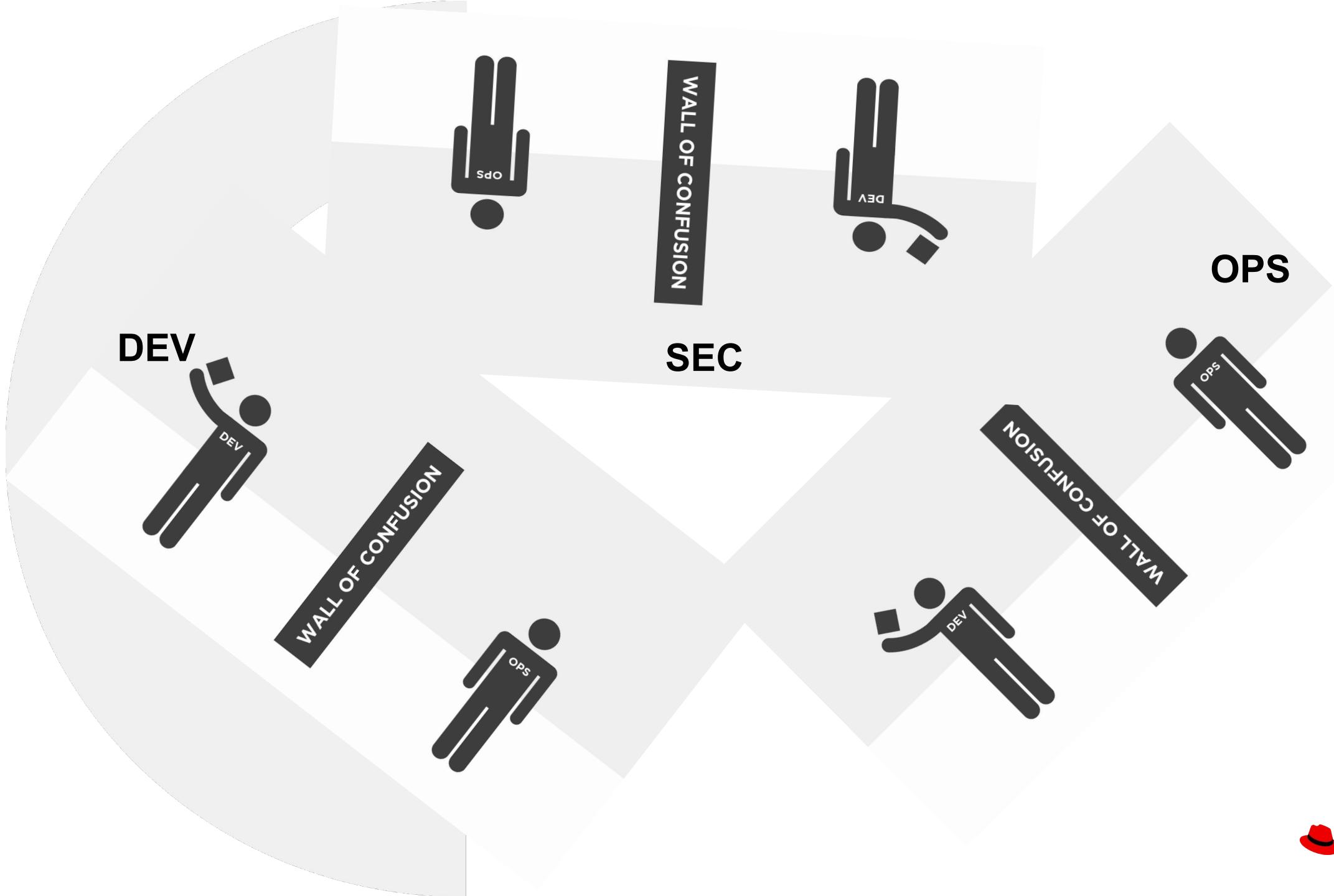
Accelerates large scale migrations and modernization with **migration toolkits**



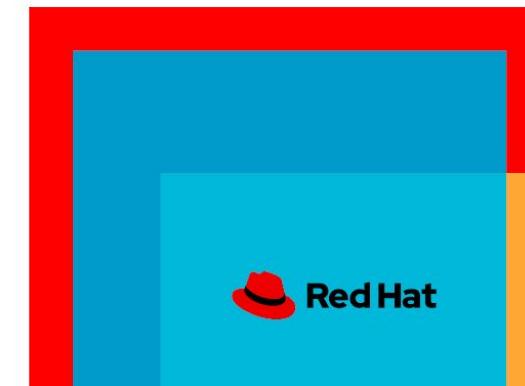
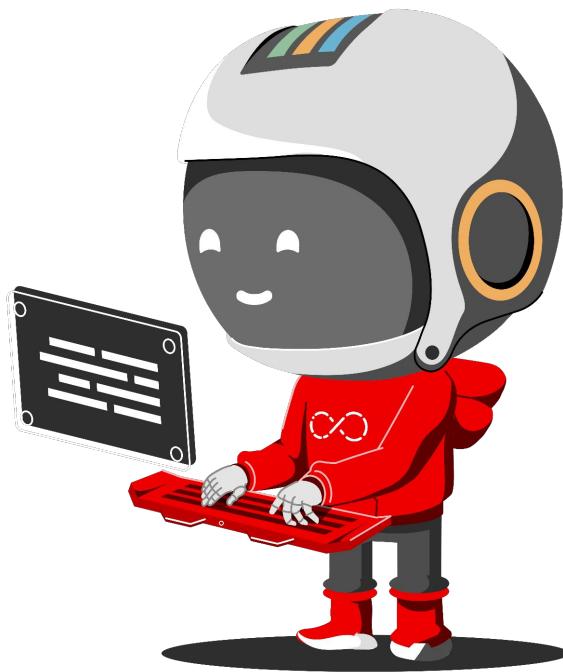
WALL OF CONFUSION







# DEMO

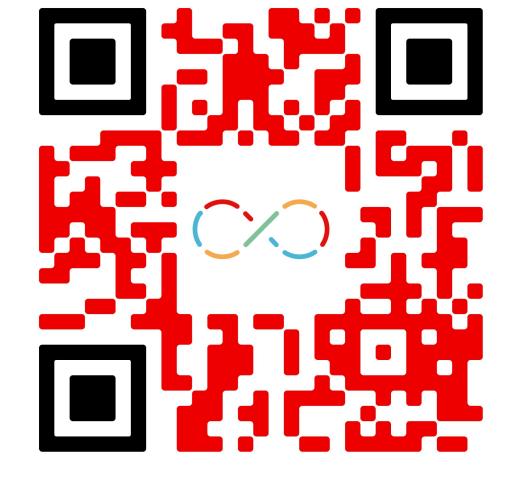




**Red Hat**

Red Hat   
Developer

Join Red Hat Developer.  
Build here. Go anywhere.



[youtube.com/RedHatDevelopers](https://youtube.com/RedHatDevelopers)



[linkedin.com/showcase/red-hat-developer](https://linkedin.com/showcase/red-hat-developer)



[facebook.com/RedHatDeveloper](https://facebook.com/RedHatDeveloper)



[twitter.com/rhdevelopers](https://twitter.com/rhdevelopers)



Connect

# Thank you



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[twitter.com/RedHat](https://twitter.com/RedHat)

