

The background features a dark gray gradient with several large, overlapping circles in yellow, orange, and blue. To the right, there are smaller spheres in blue, yellow, and red, connected by thin, curved lines, resembling a molecular or network structure.

# The Madness of Multiple Gemini CLIs Developing Simultaneously

A New World of AI Agent Collaboration Enabled by Jujutsu

**Günther Brunner**

CyberAgent, Inc.

📅 July 22, 2025 • 📍 Gemini CLI Meetup Japan #1

# About Me



**Developer Productivity &  
Performance**

Specialist



**Joined CyberAgent in 2012**



**Creator of OSS OpenSTF**

13k+ stars on GitHub



**AI Code Agents Festival Organizer**

20,000+ viewers



**Follow me on X**

@gunta85



**Jujutsu**


Next-Gen VCS

# The Problem


What if AI agents could work like human teams?




## Current Limitations

-  Working sequentially with one Gemini CLI

 One agent at a time

-  Fear of file conflicts

 Merge conflicts block progress

-  Parallel development is just a dream

 Sequential bottlenecks



## Ideal World

-  Multiple Geminis developing simultaneously

 True parallel development

-  Conflict-free collaboration

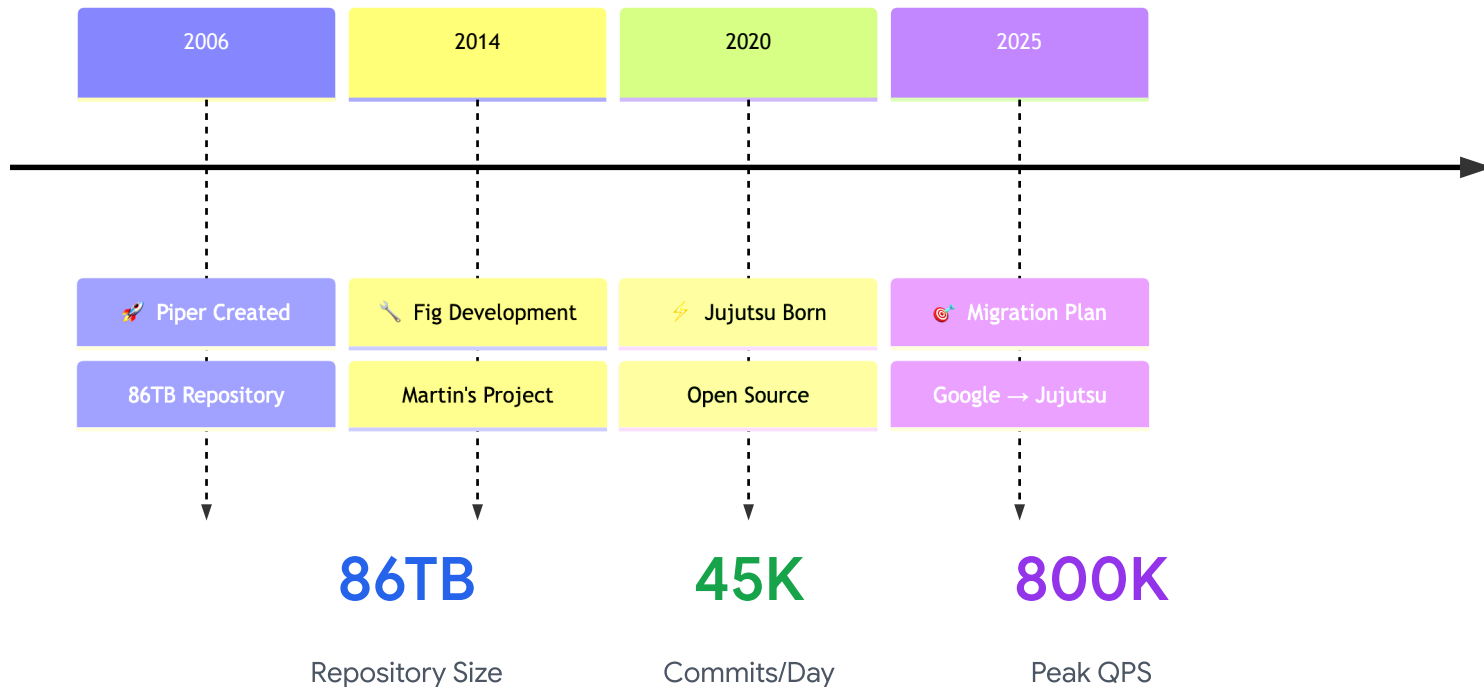
 Seamless merging

-  10x development speed

 Exponential productivity

# The Missing Piece: Google's Story

How Google solved version control at massive scale



# Jujutsu: From Google's Learnings

Next-generation VCS born from Google's experience



Martin von Zweigbergk

Creator of Jujutsu

- ✓ Git rebase co-author (2008)
- ✓ Google Fig VCS architect
- ✓ 20+ years VCS experience
- ✓ Google sponsors full-time development

## 💡 Why Jujutsu Matters

- ❤️ **First-class conflicts** - commit with conflicts
- ↩️ **Complete undo** - any operation reversible
- 📁 **True workspaces** - parallel development
- 🚀 **No staging area** - AI-friendly



Scott Chacon @chacon

GitHub Co-founder

*"If you like CLIs, Jujutsu is really interesting, and Git backended."*



Mitchell Hashimoto @mitchellh

HashiCorp Founder

*"Almost two months since I tweeted this and I've used jujutsu exclusively the entire time. I want to write something longer form but the tweet form: jj is fantastic and I can't see myself going back, only one exception is I drop down to `git` for bisect still. That's it."*

Oct 15, 2024

# The Game Changer: Conflict Sharing

## 🔗 Traditional Git Model 🔄 Jujutsu Revolution

❶ Conflicts are blockers

Work stops until resolved

❷ Must resolve before commit

Cannot save conflicted state

❸ Each worktree is isolated

No shared understanding

❶ Conflicts are data

Stored as first-class objects

❷ Commit with conflicts

Continue working immediately

❸ Workspaces share state

All agents see the same conflicts

Multiple AI agents can work on the SAME conflicted commit simultaneously



# Parallel Development Sample

3 Gemini CLIs working simultaneously



## Backend

backend-workspace

- REST API development
- Database integration



## Frontend

frontend-workspace

- React components
- UI/UX implementation



## Testing

test-workspace

- Unit tests
- Integration tests

```
# Create three workspaces
jj workspace add backend
jj workspace add frontend
jj workspace add test

# Launch Gemini CLI in each
cd backend && gemini -p "Build REST API" &
cd ../frontend && gemini -p "Create React UI" &
cd ../test && gemini -p "Write tests" &
```

# Conflict Resolution in Action

## The Conflict Scenario

```
# Three AI agents modified server.js differently:
```

**Team Auth**

app.use(auth)

**Team Database**

app.use(database)

**Team Logging**

app.use(logger)

## Git: Sequential Resolution Jujutsu: Parallel Resolution

1 Agent A hits conflict → BLOCKED

2 Agent B waits for A → BLOCKED

 **Total: 3x resolution time**

1 All agents see the conflict

2 Each works on their part

 **Total: 1x resolution time**

```
# Each agent in their workspace can:  
jj edit @ # Work on the SAME conflicted commit!
```

# Real-World Example: Feature Development

## Git Workflow

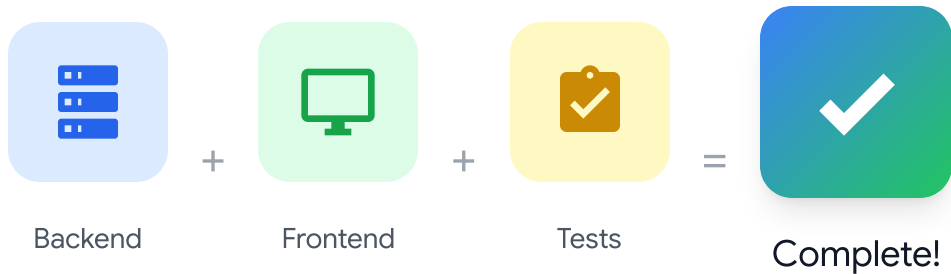
```
# Create feature branch
git checkout -b feature-payment
# Work on backend
git add backend/*
git commit -m "Add payment API"
# Switch context
git stash
git checkout -b feature-payment-ui
# Merge conflicts!
git merge feature-payment
# CONFLICT (content): Merge conflict
```

## Jujutsu Workflow

```
# Create workspace for backend
jj workspace add backend
# Create workspace for frontend
jj workspace add frontend
# Work simultaneously
cd backend && jj new
cd ../frontend && jj new
# Merge anytime
jj new @backend @frontend
# Conflicts stored, work continues!
```

# The Result

When parallel development meets coordination



Full Application in 5 Minutes

# Gemini CLI + Jujutsu: Perfect Match



## AI-Friendly Features

- ✓ No staging confusion

AI never forgets git add

- ✓ Automatic snapshots

All changes automatically saved

- ✓ Conflict-resilient

Continue working with conflicts



## Productivity Gains

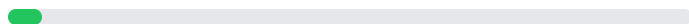
Setup Time

-90%



Conflict Resolution

-95%



Overall Speed

+1000%



# Advanced: Multi-Agent Orchestration



## Agent Network Architecture

- Coordinator Agent Task distribution
- Backend Agents (x3) API development
- Frontend Agents (x3) UI development
- Test Agents (x2) Quality assurance

## <> Sample Orchestration

```
# Coordinator creates tasks
jj workspace add coordinator
jj new -m "Project structure"

# Spawn specialized agents
for i in {1..3}; do
  jj workspace add backend-$i
  gemini -p "Build service $i" &
done

# Auto-merge all work
jj new @backend-* @frontend-*
```

# The Future at Scale

Imagine 100 Gemini CLIs working in harmony

## 100+ AI Agents



### Microservices

One AI per service

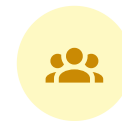
Specialized expertise



### Polyglot

Language experts

Python, Go, Rust, JS



### Specialized Roles

Security, Perf, UX

Domain experts

# Get Started Today

```
# Install Jujutsu
brew install jj
# Initialize repository
jj init --git
# Create workspaces
jj workspace add feature-a
jj workspace add feature-b
# Launch Gemini CLI
gemini -p "Build amazing features"
```



## Documentation

Official Jujutsu docs

[github.com/jj-vcs/jj](https://github.com/jj-vcs/jj)



## Tutorial

Interactive learning

[jj-vcs.github.io](https://jj-vcs.github.io)



# Thank You

Günther Brunner

CyberAgent, Inc.

✕ @gunta85

Let's Build the Future Together