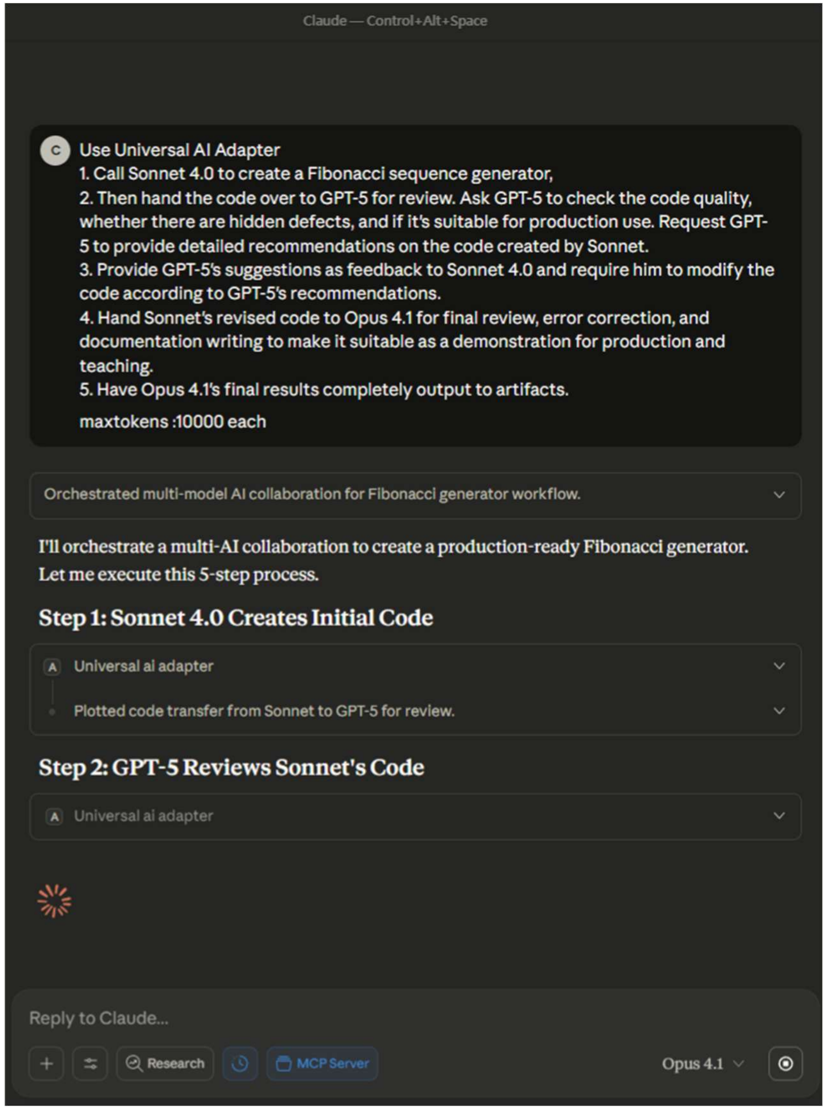
AI Collaboration among Sonnet, GPT-5, Opus in Claude Windows



Use Universal AI Adapter
1. Call Sonnet 4.0 to create a Fibonacci sequence generator,
2. Then hand the code over to GPT-5 for review. Ask GPT-5 to check the code quality, whether there are hidden defects, and if it's suitable for production use. Request GPT-5 to provide detailed recommendations on the code created by Sonnet.
3. Provide GPT-5's suggestions as feedback to Sonnet 4.0 and require him to modify the code according to GPT-5's recommendations.
4. Hand Sonnet's revised code to Opus 4.1 for final review, error correction, and documentation writing to make it suitable as a demonstration for production and teaching.
5. Have Opus 4.1's final results completely output to artifacts.

maxtokens :10000 each

Orchestrated multi-model AI collaboration for Fibonacci generator workflow.

I'll orchestrate a multi-AI collaboration to create a production-ready Fibonacci generator. Let me execute this 5-step process.

Step 1: Sonnet 4.0 Creates Initial Code

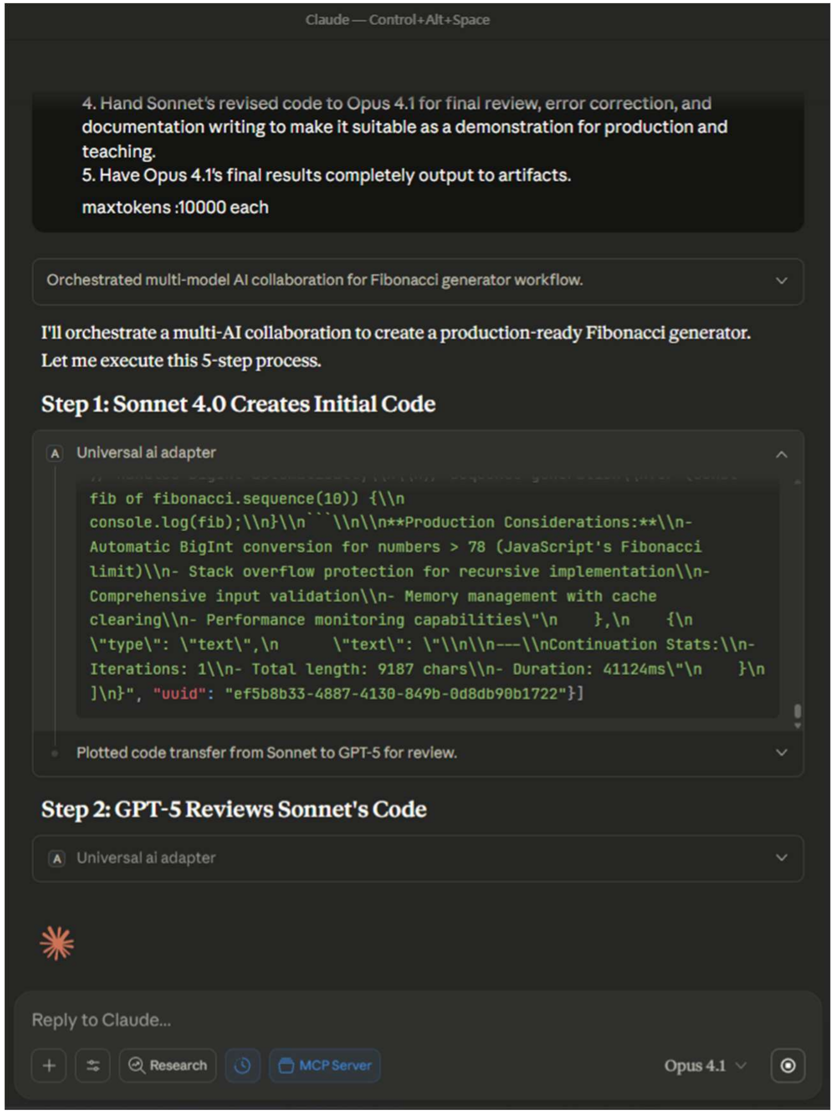
Universal ai adapter

Plotted code transfer from Sonnet to GPT-5 for review.

Step 2: GPT-5 Reviews Sonnet's Code

Universal ai adapter

Step 1. Sonnet 4.0 API create code upon request



4. Hand Sonnet's revised code to Opus 4.1 for final review, error correction, and documentation writing to make it suitable as a demonstration for production and teaching.
5. Have Opus 4.1's final results completely output to artifacts.

maxtokens :10000 each

Orchestrated multi-model AI collaboration for Fibonacci generator workflow.

I'll orchestrate a multi-AI collaboration to create a production-ready Fibonacci generator. Let me execute this 5-step process.

Step 1: Sonnet 4.0 Creates Initial Code

Universal ai adapter

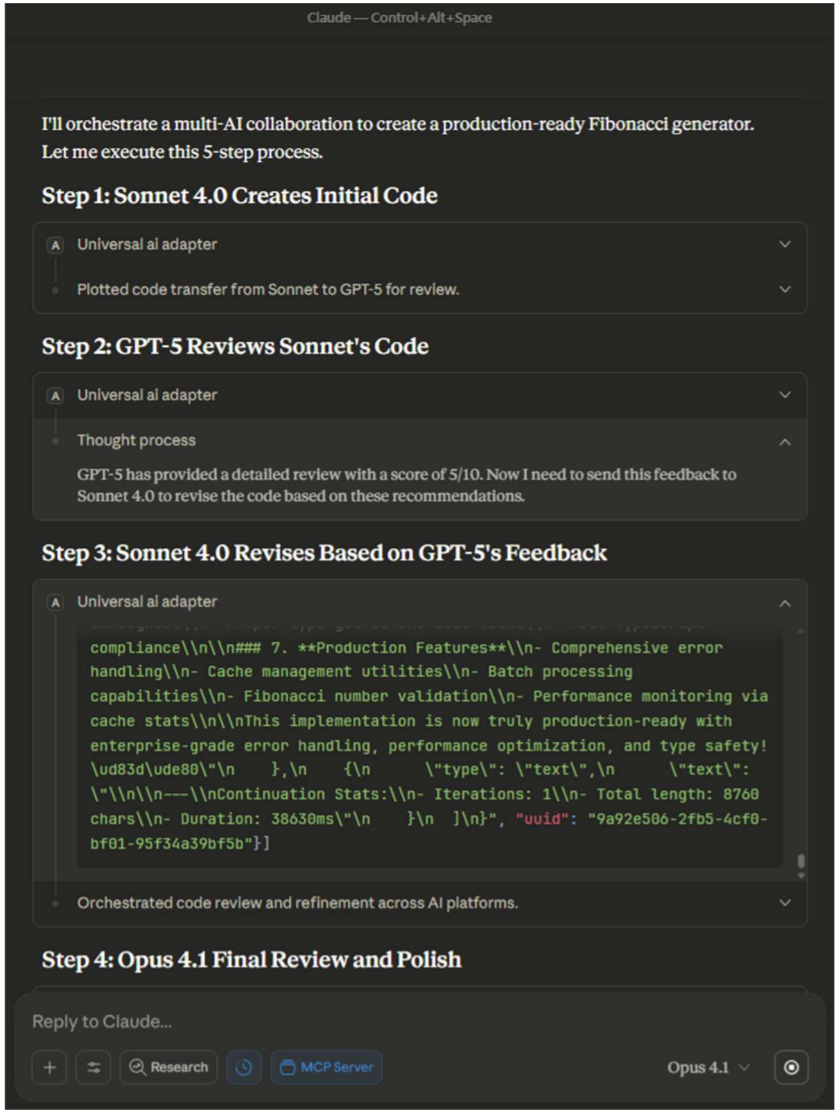
```
fib of fibonacci.sequence(10)) {\\n
console.log(fib);\\n}\\n```\\n\\n**Production Considerations:**\\n-
Automatic BigInt conversion for numbers > 78 (JavaScript's Fibonacci
limit)\\n- Stack overflow protection for recursive implementation\\n-
Comprehensive input validation\\n- Memory management with cache
clearing\\n- Performance monitoring capabilities\"\n    },\n    {\n
\"type\": \"text\",\n      \"text\": \"\\n\\n---\\nContinuation Stats:\\n-
Iterations: 1\\n- Total length: 9187 chars\\n- Duration: 41124ms\"\n    }\n
]\n}", "uuid": "ef5b8b33-4887-4130-849b-0d8db90b1722"}]
```

Plotted code transfer from Sonnet to GPT-5 for review.

Step 2: GPT-5 Reviews Sonnet's Code

Universal ai adapter

Step 2. GPT-5 API review Sonnet's code



I'll orchestrate a multi-AI collaboration to create a production-ready Fibonacci generator. Let me execute this 5-step process.

Step 1: Sonnet 4.0 Creates Initial Code

Universal ai adapter

Plotted code transfer from Sonnet to GPT-5 for review.

Step 2: GPT-5 Reviews Sonnet's Code

Universal ai adapter

Thought process

GPT-5 has provided a detailed review with a score of 5/10. Now I need to send this feedback to Sonnet 4.0 to revise the code based on these recommendations.

Step 3: Sonnet 4.0 Revises Based on GPT-5's Feedback

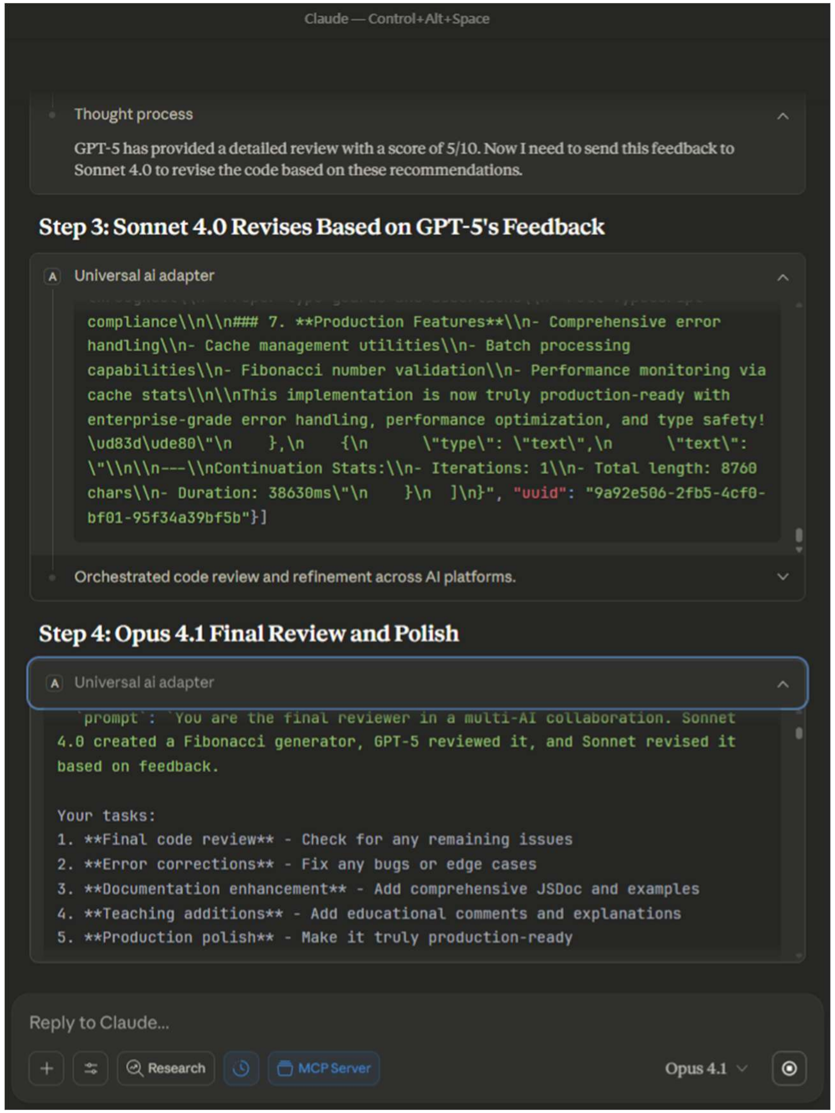
Universal ai adapter

```
compliance\\n\\n### 7. **Production Features**\\n- Comprehensive error
handling\\n- Cache management utilities\\n- Batch processing
capabilities\\n- Fibonacci number validation\\n- Performance monitoring via
cache stats\\n\\nThis implementation is now truly production-ready with
enterprise-grade error handling, performance optimization, and type safety!
\ud83d\ude80\"\n    },\n    {\n      \"type\": \"text\",\n      \"text\":
\"\\n\\n---\\nContinuation Stats:\\n- Iterations: 1\\n- Total length: 8760
chars\\n- Duration: 38630ms\"\n    }\n  ]\n}", "uuid": "9a92e506-2fb5-4cf0-
bf01-95f34a39bf5b"}]
```

Orchestrated code review and refinement across AI platforms.

Step 4: Opus 4.1 Final Review and Polish

Step 3. Sonnet revises based on GPT-5 feedback



Thought process

GPT-5 has provided a detailed review with a score of 5/10. Now I need to send this feedback to Sonnet 4.0 to revise the code based on these recommendations.

Step 3: Sonnet 4.0 Revises Based on GPT-5's Feedback

Universal ai adapter

```
compliance\\n\\n### 7. **Production Features**\\n- Comprehensive error
handling\\n- Cache management utilities\\n- Batch processing
capabilities\\n- Fibonacci number validation\\n- Performance monitoring via
cache stats\\n\\nThis implementation is now truly production-ready with
enterprise-grade error handling, performance optimization, and type safety!
\ud83d\ude80\"\n    },\n    {\n      \"type\": \"text\",\n      \"text\":
\"\\n\\n---\\nContinuation Stats:\\n- Iterations: 1\\n- Total length: 8760
chars\\n- Duration: 38630ms\"\n    }\n  ]\n}", "uuid": "9a92e506-2fb5-4cf0-
bf01-95f34a39bf5b"}]
```

Orchestrated code review and refinement across AI platforms.

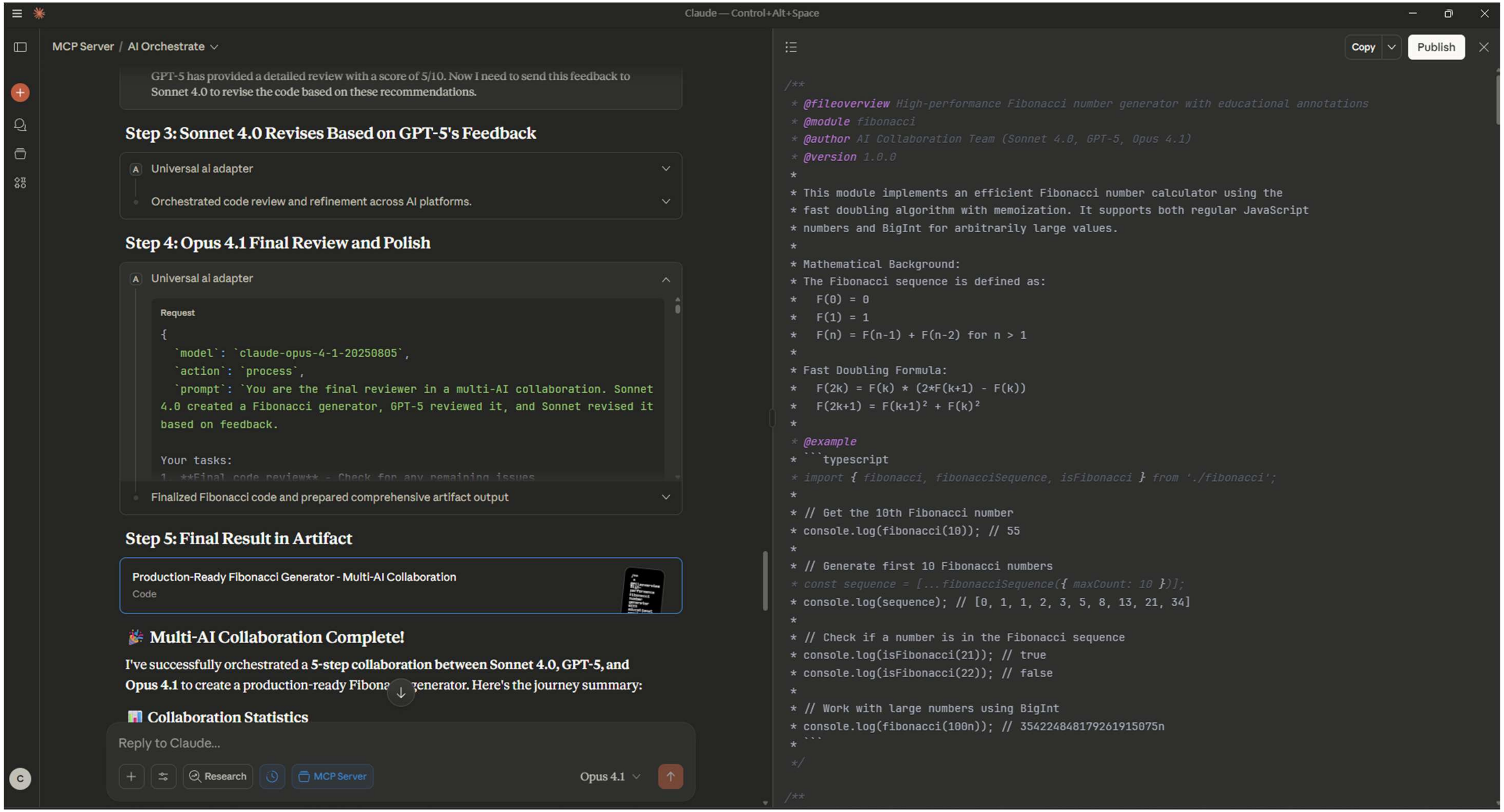
Step 4: Opus 4.1 Final Review and Polish

Universal ai adapter

```
  `prompt`: `You are the final reviewer in a multi-AI collaboration. Sonnet
4.0 created a Fibonacci generator, GPT-5 reviewed it, and Sonnet revised it
based on feedback.

Your tasks:
1. **Final code review** - Check for any remaining issues
2. **Error corrections** - Fix any bugs or edge cases
3. **Documentation enhancement** - Add comprehensive JSDoc and examples
4. **Teaching additions** - Add educational comments and explanations
5. **Production polish** - Make it truly production-ready
```

Step 4. Opus 4.1 API final review and polish



GPT-5 has provided a detailed review with a score of 5/10. Now I need to send this feedback to Sonnet 4.0 to revise the code based on these recommendations.

Step 3: Sonnet 4.0 Revises Based on GPT-5's Feedback

Universal ai adapter

Orchestrated code review and refinement across AI platforms.

Step 4: Opus 4.1 Final Review and Polish

Universal ai adapter

Request

```
{
  `model`: `claude-opus-4-1-20250805`,
  `action`: `process`,
  `prompt`: `You are the final reviewer in a multi-AI collaboration. Sonnet 4.0 created a Fibonacci generator, GPT-5 reviewed it, and Sonnet revised it based on feedback.

Your tasks:
1. **Final code review** - Check for any remaining issues
```

Finalized Fibonacci code and prepared comprehensive artifact output

Step 5: Final Result in Artifact

Production-Ready Fibonacci Generator - Multi-AI Collaboration
Code

🎉 Multi-AI Collaboration Complete!

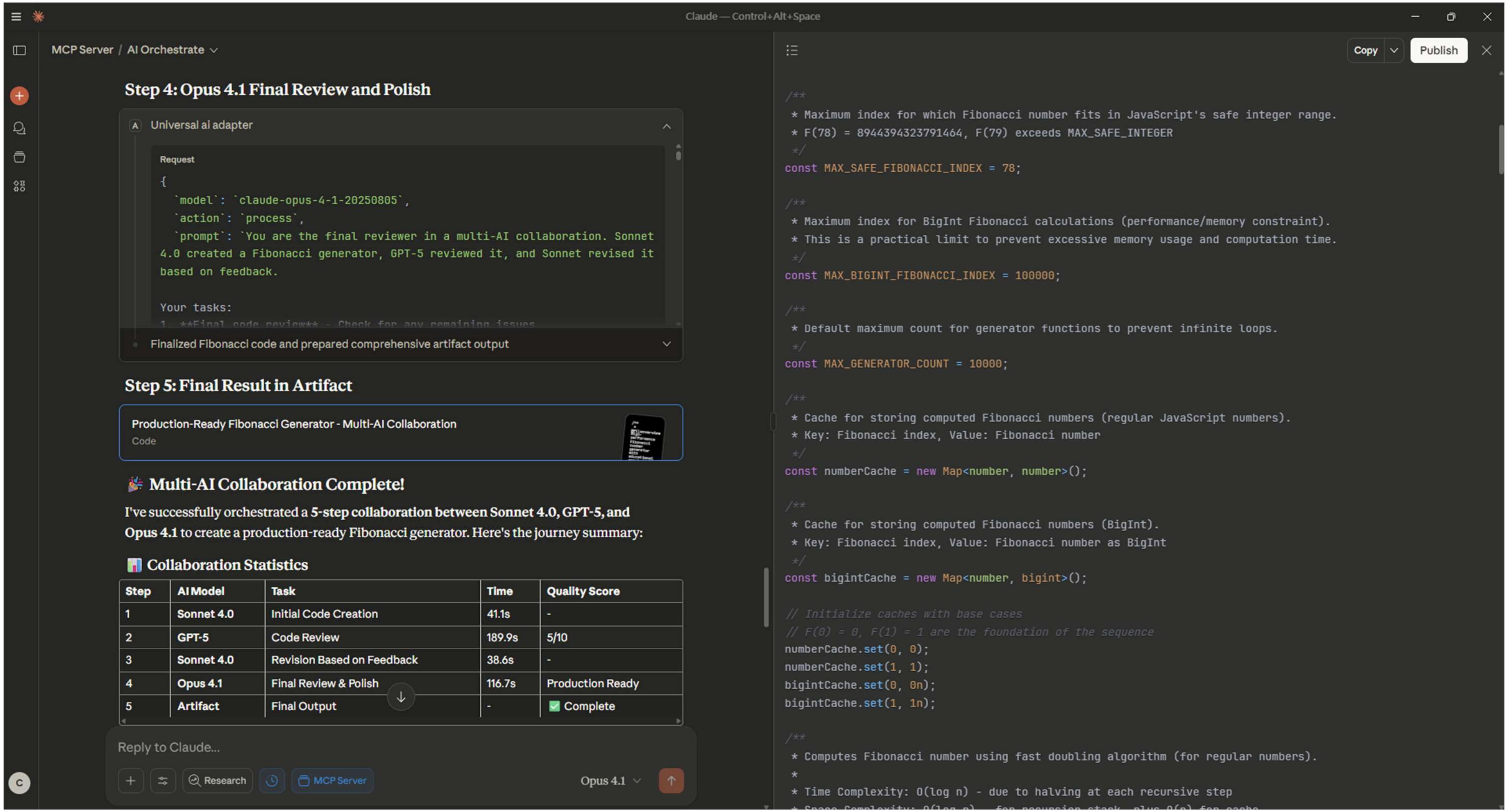
I've successfully orchestrated a **5-step collaboration between Sonnet 4.0, GPT-5, and Opus 4.1** to create a production-ready Fibona generator. Here's the journey summary:

📊 Collaboration Statistics

````
/**
 * @fileoverview High-performance Fibonacci number generator with educational annotations
 * @module fibonacci
 * @author AI Collaboration Team (Sonnet 4.0, GPT-5, Opus 4.1)
 * @version 1.0.0
 *
 * This module implements an efficient Fibonacci number calculator using the
 * fast doubling algorithm with memoization. It supports both regular JavaScript
 * numbers and BigInt for arbitrarily large values.
 *
 * Mathematical Background:
 * The Fibonacci sequence is defined as:
 *   F(0) = 0
 *   F(1) = 1
 *   F(n) = F(n-1) + F(n-2) for n > 1
 *
 * Fast Doubling Formula:
 *   F(2k) = F(k) * (2*F(k+1) - F(k))
 *   F(2k+1) = F(k+1)² + F(k)²
 *
 * @example
 * ```typescript
 * import { fibonacci, fibonacciSequence, isFibonacci } from './fibonacci';
 *
 * // Get the 10th Fibonacci number
 * console.log(fibonacci(10)); // 55
 *
 * // Generate first 10 Fibonacci numbers
 * const sequence = [...fibonacciSequence({ maxCount: 10 })];
 * console.log(sequence); // [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
 *
 * // Check if a number is in the Fibonacci sequence
 * console.log(isFibonacci(21)); // true
 * console.log(isFibonacci(22)); // false
 *
 * // Work with large numbers using BigInt
 * console.log(fibonacci(100n)); // 354224848179261915075n
 * ```
 */

/**
````

Step 5. Final work of Sonnet 4.0, GPT-5, Opus 4.1 in Artifacts



Step 4: Opus 4.1 Final Review and Polish

Universal ai adapter

Request

```
{
  `model`: `claude-opus-4-1-20250805`,
  `action`: `process`,
  `prompt`: `You are the final reviewer in a multi-AI collaboration. Sonnet
4.0 created a Fibonacci generator, GPT-5 reviewed it, and Sonnet revised it
based on feedback.

Your tasks:
1. **Final code review** - Check for any remaining issues
```

Finalized Fibonacci code and prepared comprehensive artifact output

Step 5: Final Result in Artifact

Production-Ready Fibonacci Generator - Multi-AI Collaboration
Code

🎉 Multi-AI Collaboration Complete!

I've successfully orchestrated a **5-step collaboration between Sonnet 4.0, GPT-5, and Opus 4.1** to create a production-ready Fibonacci generator. Here's the journey summary:

📊 Collaboration Statistics

Step	AI Model	Task	Time	Quality Score
1	**Sonnet 4.0**	Initial Code Creation	41.1s	-
2	**GPT-5**	Code Review	189.9s	5/10
3	**Sonnet 4.0**	Revision Based on Feedback	38.6s	-
4	**Opus 4.1**	Final Review & Polish	116.7s	Production Ready
5	**Artifact**	Final Output	-	✅ Complete

```
/**
 * Maximum index for which Fibonacci number fits in JavaScript's safe integer range.
 * F(78) = 8944394323791464, F(79) exceeds MAX_SAFE_INTEGER
 */
const MAX_SAFE_FIBONACCI_INDEX = 78;

/**
 * Maximum index for BigInt Fibonacci calculations (performance/memory constraint).
 * This is a practical limit to prevent excessive memory usage and computation time.
 */
const MAX_BIGINT_FIBONACCI_INDEX = 100000;

/**
 * Default maximum count for generator functions to prevent infinite loops.
 */
const MAX_GENERATOR_COUNT = 10000;

/**
 * Cache for storing computed Fibonacci numbers (regular JavaScript numbers).
 * Key: Fibonacci index, Value: Fibonacci number
 */
const numberCache = new Map<number, number>();

/**
 * Cache for storing computed Fibonacci numbers (BigInt).
 * Key: Fibonacci index, Value: Fibonacci number as BigInt
 */
const bigintCache = new Map<number, bigint>();

// Initialize caches with base cases
// F(0) = 0, F(1) = 1 are the foundation of the sequence
numberCache.set(0, 0);
numberCache.set(1, 1);
bigintCache.set(0, 0n);
bigintCache.set(1, 1n);

/**
 * Computes Fibonacci number using fast doubling algorithm (for regular numbers).
 *
 * Time Complexity: O(log n) - due to halving at each recursive step
 * Space Complexity: O(log n) - for recursion stack, plus O(n) for cache
```

AI Collaboration: Production-Grade Code Completed in 6.4 Minutes