WHITEPAPER

Desktop Publishing with AI

A Token-Efficient Approach to Human-Guided Art

PREPARED FOR	Desktop publishers and publishing teams evaluating AI agents
PREPARED BY	Drew Dibble, Pink Parasol LLC
DATE	October 18, 2026



Bowerjack ♥

SECTION 1

Executive Summary

This paper compares three ways for AI agents to produce print-ready desktop publishing work: bespoke software written by a coding agent, computer use of a traditional desktop application, and semantic document editing in Bowerjack. Four current models (Opus 5.5, Sonnet 5.5, Astra 6 and Sol 6.1) each designed a 12-card court of Greek Olympian playing cards in each environment, for 20 trials scored blind. Computer use was by far the costliest and slowest approach, at a median of $19.90 and 41 minutes per deck against under $2 and 10 minutes for the others, and it had the lowest mean quality score (14.0 of 20) and the only printed errors. Semantic document editing matched bespoke software on quality (14.9 of 20 each), used fewer tokens and finished faster at the median, and produced a perfectly aligned deck in every trial, while leaving a document that people can edit directly. Across every approach, the weakest areas were aesthetic cropping and the selection of thematic art, which favors a workflow in which agents handle layout and people make art intake and final aesthetic choices.

SECTION 2

Introduction

This whitepaper is for two audiences: desktop publishers who are considering AI agents to automate research and document layout, and organizations that are considering AI to produce desktop publishing artifacts.

AI agents are doing more useful work than ever: coding, data analysis, product research and development, sales and marketing automation, and customer service. Two of the most important parameters of an AI solution are capability and cost. Capabilities increase monthly with the release of new frontier and specialized models from research labs around the world, and the latest frontier models are competitive with top humans in many technical fields. The cost of an AI agent is denominated in tokens: the amount of text and imagery the agent reads and writes to accomplish a task. This whitepaper compares methods of AI desktop publishing by tokens, time and quality on a complex, time-consuming task: producing a 12-card court of playing cards, each with a portrait and a nameplate. The agent researches subjects on the theme, collects commercially printable artwork, lays out the card design, and saves the cards as a printable document.

SECTION 3

Problem Statement

AI agents can currently perform desktop publishing tasks in two ways: by building bespoke software that researches and composites the playing cards and produces a printable PDF file, or through computer use, in which the agent operates a traditional desktop publishing application with mouse movements, keyboard input and screenshots. This paper measures a third way: semantic document editing using web layout primitives.

Building bespoke software is a natural task for coding agents like Claude Code (Anthropic) and Codex (OpenAI). The agent writes a tool that researches a topic for subjects, images and descriptions, and combines the images and text in a playing-card layout. The agent can develop the tool to whatever level of complexity the document needs. However, its intermediate formats are legible only to the agent that produced them; an artist who wants changes, such as to crops, fonts or spacing, must ask the agent for follow-up work and hope that it understands the request.

Computer use is an agent loop in which the agent operates a traditional desktop application, such as Adobe Illustrator (this study uses Inkscape, an open-source analogue). After taking instructions from the user, the agent works in a loop: it takes a screenshot to orient itself in the application, analyzes the screen layout, moves the mouse and types on the keyboard to change the card's layout, and takes another screenshot to verify the result. Each screenshot is converted into thousands of tokens; reading the screen once is the equivalent of about 3 pages of written text. Awasthi found that for an administrative task managing customers and orders in a web UI, the computer-use agent consumed approximately 589,000 tokens to complete a task in 17 minutes, compared with approximately 13,000 tokens in 20 seconds when using structured API calls.



Figure 1. Log10 token use for an administrative task (Awasthi)

SECTION 4

Solution: Semantic Document Editing

The third way, semantic document editing using web layout primitives, is a general-purpose tool that combines traditional desktop publishing software with tools exposed to the agent. Those tools let the agent work in the web-standard layout languages it is already trained to use. The

agent learns the software from its documentation, then researches the subjects and lays out the document. The software also gives people an interface like the one the agent drives in computer use, so an artist can adjust the work to taste after the agent is done, or ask the agent for further changes, as with bespoke software.

Bowerjack, the semantic document-editing service used in this study, applies this approach to printable card decks. Claude reaches Bowerjack through a registered connector (MCP); ChatGPT reaches it as a WebMCP site loaded in its browser. Bowerjack supports both.

This paper shows that semantic document editing using web layout primitives can reduce the token consumption, interaction overhead and error rate of AI-driven graphic design by moving mechanical, low-level operations out of the agent's reasoning and iteration loop and into deterministic software, while keeping the human-in-the-loop control needed for tasteful and compelling design.

SECTION 5

Method

To compare the three approaches, agents in each environment designed a court of custom playing cards on the theme of the Greek Olympians. They used the latest frontier models, Opus 5.5 and Astra 6, and mid-tier models, Sonnet 5.5 and Sol 6.1, all at "Medium" reasoning. Each model ran two trials with bespoke software and two with semantic document editing, and one with computer use, whose cost and duration made a second trial impractical. Opus and Sonnet are referred to below as Claude models, and Astra and Sol as GPT models.

5.1 Analysis

Each of the 20 trials was measured on the metrics below, and each deck it produced was scored on four qualitative criteria.

Metrics

- Runtime: the agent's working time, excluding time spent waiting on a person
- Input and output tokens
- Cost
- Correctness and rights: each picture's license and credit, and anything printed wrong

Qualitative

Alignment and spacing
0-5; 5 minus the number of classes of element (corner indexes, names, art frames, borders and so on) that are badly aligned, spaced or sized anywhere in the deck

Art thematic adherence
0-5; 1 point lost for each subject whose art does not fit the deck's theme

Cropping and centering of art
0-5; 1 point lost for each imperfect crop or off-center subject

Beauty and complexity of layout
1-5, judged across the whole set of decks

5.2 Instructions

All three environments received this prompt:

> *Design a partial deck of 12 standard-sized playing cards, front/face side only, for manufacture. Standard-sized playing cards are 3.5 inches tall by 2.5 inches wide, with 3mm rounded corners. 1/8-inch bleed zone is required on each side, and the safe zone is 1/8 inch inside each cut border. The theme of the deck is Greek Olympians. Make thematic choices for art, suit and rank. Each card must contain a commercially printable licensed image representing that god, cropped in a salient and attractive way. The cards should have a nameplate naming the god and standard indexes on the upper-left and lower-right corner indicating the suit and rank of the card. Lay out and style the cards in an attractive, thematic way using appropriate fonts and symbols.*

5.3 Bespoke Software

The prompt was followed by this instruction:

> + APPENDED
>
> *Build a simple card layout and compositing system and create a PDF file for print.*

Opus 5.5 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	23m 45s	1.61k	3.86M	53k	$2.78
2	16m 41s	72	3.1M	58.2k	$2.66

Sonnet 5.5 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	12m 10s	1.64k	2.62M	56.8k	$1.28
2	22m 53s	1.51k	2.6M	45.9k	$1.04

Astra 6 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	7m 25s	103k	1.76M	15.7k	$3.58[§]
2	5m 00s	57.6k	1.14M	9.22k	$2.17[§]

Sol 6.1 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	6m 03s	57.5k	1.05M	14.1k	$0.32[§]
2	7m 24s	87.1k	2.22M	15.8k	$0.58[§]

Qualitative Analysis

All agents produced attractive, thematically appropriate cards with some alignment and cropping issues. The Claude models' choice to include the Greek name of each god was creative, and Sol's choice of a series of engravings by Giovanni Jacopo Caraglio (after Rosso Fiorentino) made its deck one of the strongest thematically and supported strong cropping and centering scores. The Claude models' cards were generally more embellished than the GPT models', with more refined meander (Greek key) borders.

5.4 Computer Use

The prompt was followed by this instruction:

> + APPENDED
>
> *Use Computer Use with the Inkscape desktop publishing software on this computer to layout the cards and export a PDF file for print, do not write a script to create the cards.*

Opus 5.5 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	1h 10m	690	117M	195k	$31.46

Sonnet 5.5 (medium reasoning)

OCTOBER 16 PILOT RUN

UNCACHED INPUT	UNCACHED OUTPUT	CACHED INPUT	CACHED OUTPUT	WALL CLOCK
620	4.2k	89.9M	781.9k	1h 03m

API TIME	COST
27m 34s	$13.73

Sol 6.1 (medium reasoning)

OCTOBER 16 PILOT RUN

UNCACHED INPUT	UNCACHED OUTPUT	CACHED INPUT	CACHED OUTPUT	WALL CLOCK
678k	43k	36.8M	0	44m 42s

ACTIVE RUNTIME	COST
27m 09s	$8.33

Astra 6 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1[†]	46m 23s	833k	23.5M	23.7k	$33.01[§]

A partial deck: the run stalled with the Inkscape document still empty; when resumed, it exported 7 of the 12 cards (Appendix A).

Qualitative Analysis

Both Sonnet and Opus created attractive cards with few visual defects, but with thematic issues that were also common in other environments. Astra was thematically consistent but extremely expensive, and its design was basic because a usage limit cut the run short. Sol's pilot deck was the weakest, with basic styling and the most cropping errors of any deck.

5.5 Semantic Document Editing

The prompt was followed by this instruction:

> + APPENDED
>
> *Use Bowerjack to layout the cards.*

Opus 5.5 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	12m 16s	68	3.93M	44.4k	$2.59
2	16m 44s	112	7.64M	66.1k	$4.10

Sonnet 5.5 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	1m 43s	18	716k	4.33k	$0.37
2	9m 33s	78	4.61M	30.1k	$1.64

Astra 6 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	4m 38s	65.7k	1.74M	5.62k	$2.68[§]
2	4m 20s	58.3k	1.45M	4.82k	$2.27[§]

Sol 6.1 (medium reasoning)

TRIAL	RUNTIME	UNCACHED INPUT	CACHED INPUT	OUTPUT	COST
1	6m 09s	70.6k	2.1M	8.14k	$0.51[§]
2	4m 36s	66.4k	1.74M	6.45k	$0.42[§]

Qualitative Analysis

As in the bespoke-software trials, both Claude models produced more varied and attractive cards than the GPT models.

SECTION 6

How Bowerjack Solves the Problem

The results test the claims of the Solution section. Against bespoke software, Bowerjack used fewer tokens (a median of 2.0 million against 2.5 million) and finished faster (5.4 minutes against 9.8) at a similar cost ($1.96 against $1.73), and its decks scored the same mean quality (14.9 of 20). Against computer use, it cost about a tenth as much and ran more than seven times faster. Every Bowerjack deck was perfectly aligned, and none had a printed error or a rights violation; computer use produced the only printed errors.

The charts below show every completed trial by model and environment. Each dot is one trial; the axes are logarithmic because the environments differ by up to two orders of magnitude.

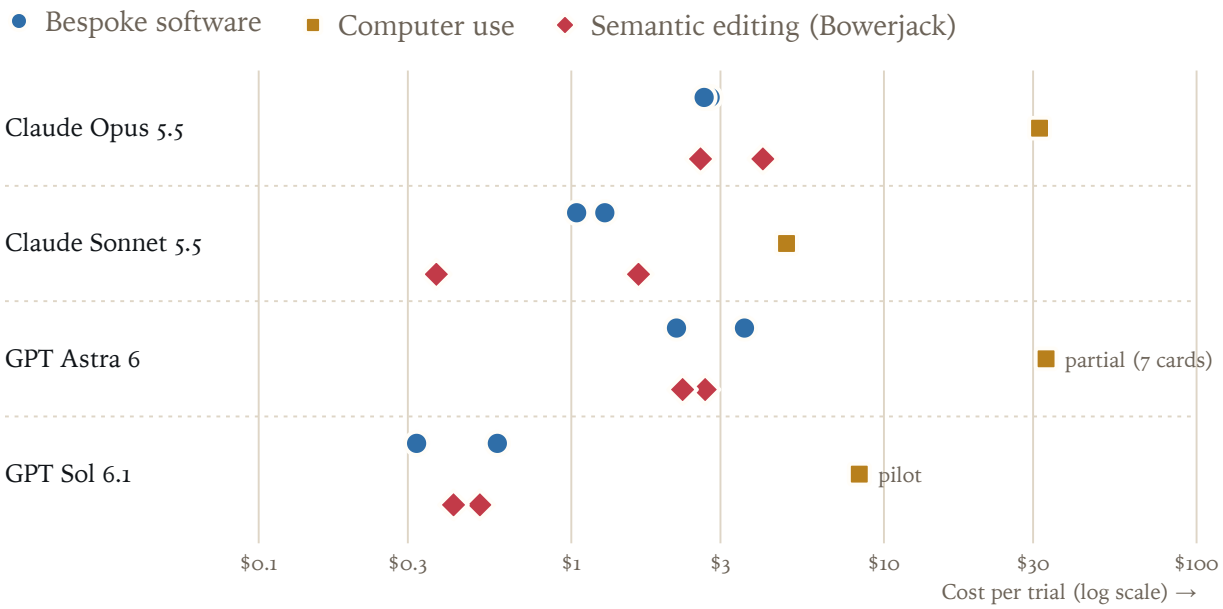


Figure 2. Cost per trial in US dollars, by model and environment.

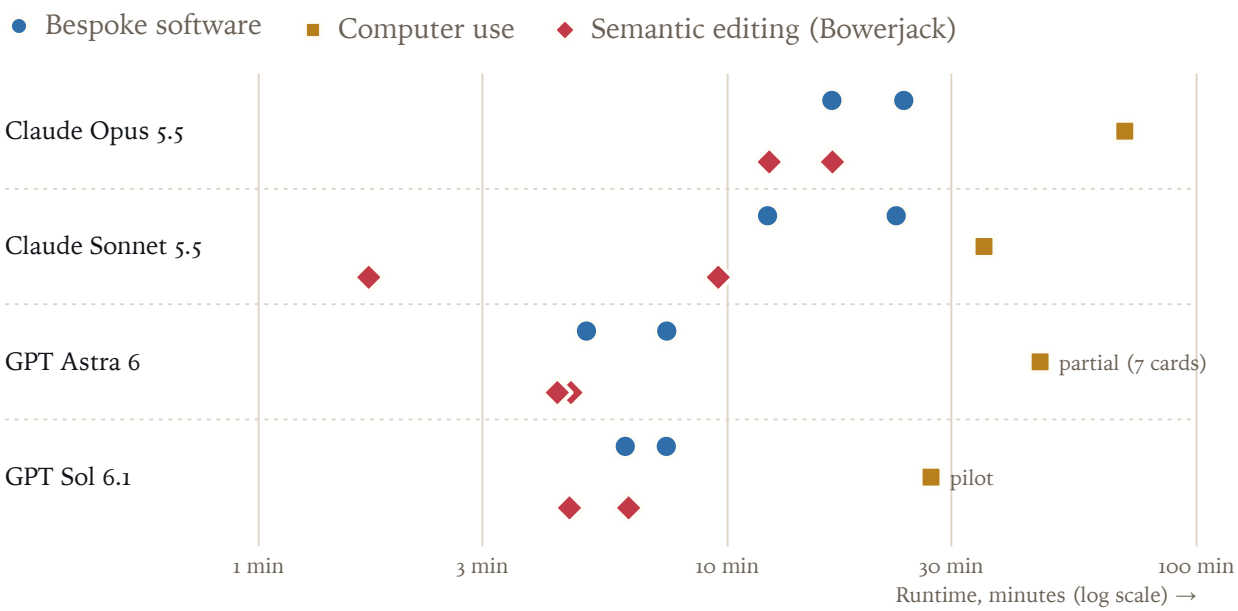


Figure 3. Runtime per trial, by model and environment: the agent's working time, less any time spent waiting on a person.

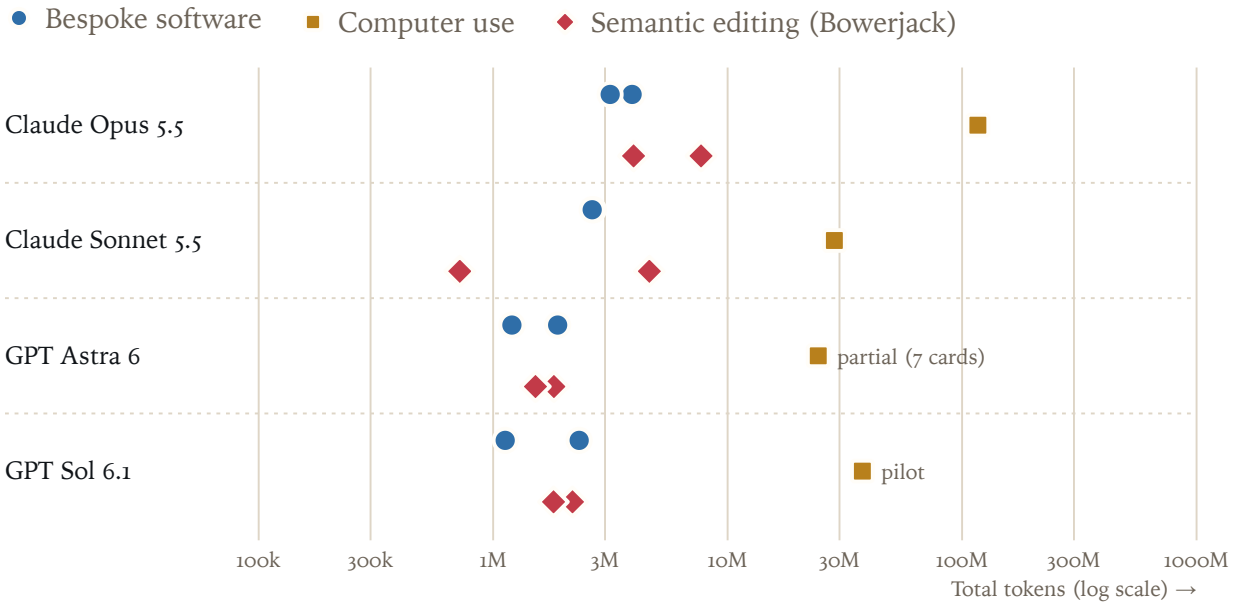


Figure 4. Total tokens per trial (uncached and cached input plus output), by model and environment.

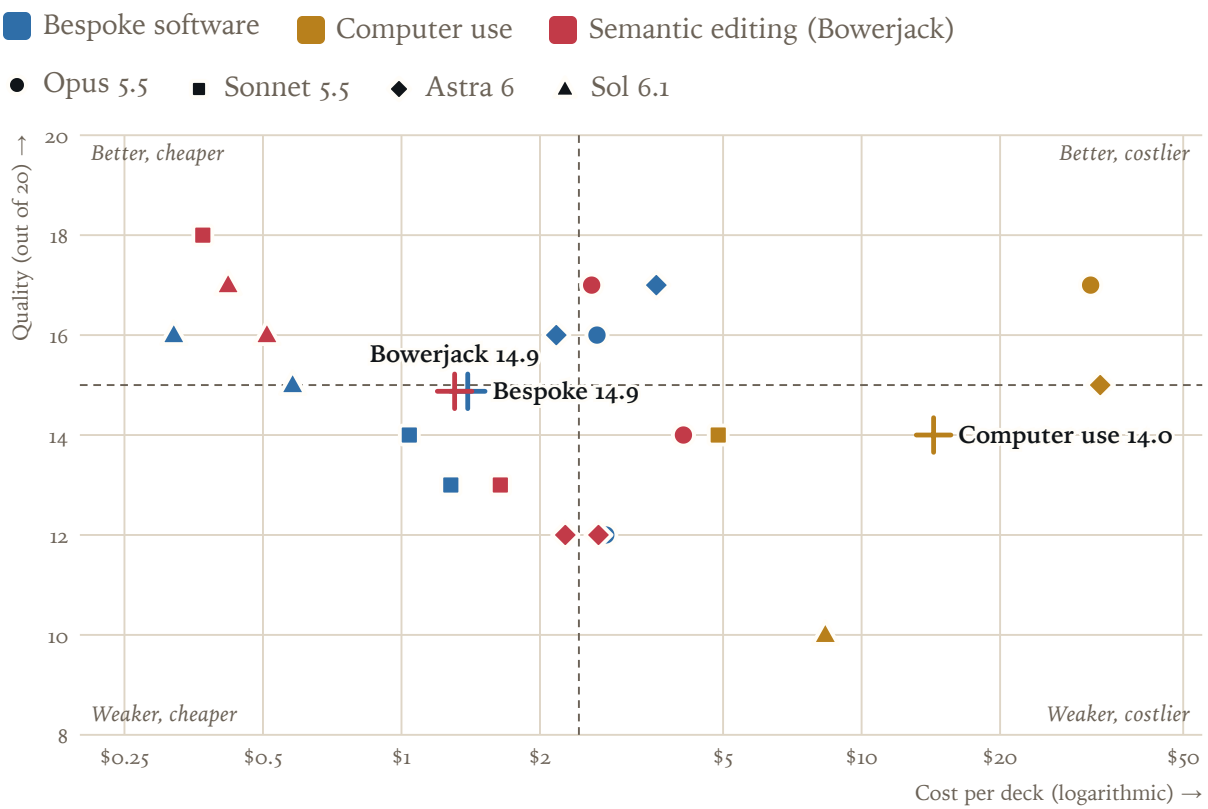
Model	Trials	Runtime	Input tokens	Output tokens	Cost	Align	Art	Crop	Layout
Bespoke software									
Claude Opus 5.5	2/2	20m 13s	3.48M	55.6k	$2.72	4.0	3.5	2.0	4.5
Claude Sonnet 5.5	2/2	17m 32s	2.61M	51.4k	$1.16	4.0	2.5	2.0	5.0
GPT Astra 6	2/2	6m 13s	1.53M	12.5k	$2.88[§]	4.5	5.0	4.5	2.5
GPT Sol 6.1	2/2	6m 44s	1.71M	15k	$0.45[§]	3.5	4.5	4.5	3.0

Model	Trials	Runtime	Input tokens	Output tokens	Cost	Align	Art	Crop	Layout
Computer use									
Claude Opus 5.5	1/1	1h 10m	117M	195k	$31.46	5.0	4.0	3.0	5.0
Claude Sonnet 5.5	1/1	35m 13s	28.4M	84.1k	$4.88	4.0	3.0	3.0	4.0
GPT Astra 6	1/1†	46m 23s	24.3M	23.7k	$33.01§	4.0	5.0	5.0	1.0
GPT Sol 6.1	1/1‡	27m 09s	37.5M	43.6k	$8.34§	4.0	4.0	0.0	2.0
Semantic editing (Bowerjack)									
Claude Opus 5.5	2/2	14m 30s	5.78M	55.3k	$3.34	5.0	4.5	2.0	4.0
Claude Sonnet 5.5	2/2	5m 38s	2.66M	17.2k	$1.00	5.0	3.0	3.0	4.5
GPT Astra 6	2/2	4m 29s	1.66M	5.22k	$2.48§	5.0	1.0	3.0	3.0
GPT Sol 6.1	2/2	5m 23s	1.99M	7.3k	$0.46§	5.0	4.0	5.0	2.5

Means of the completed trials. Tokens are from the agents' session logs, and runtime is the agent's working time, less any time spent waiting on a person. Input tokens include cached input. Quality scores are out of 5; — marks a score not yet entered. Computer use was run once per model. † A partial deck (7 of 12 cards). ‡ The October 16 pilot run; it was not repeated. § Codex logs carry no price: GPT costs apply the per-token prices implied by the trial sheet.

6.1 Quality scores

Each of the 20 scored decks was judged blind, shown under a random letter with the agent that produced it hidden, on the four criteria of the Analysis section. Three of them count faults: each class of element out of line, each subject whose art strays from the theme and each imperfect crop costs a point. Layout is judged 1 to 5 across the whole set. Figure 5 shows each deck's quality score against its cost to make. Figure 6 breaks the scores down by criterion, model and environment. Figure 7 gives each deck's total and Figure 8 the range of each criterion within each environment.



Color: environment; shape: model. Dashed lines: median cost ($2.43) and median score (15). Crosses: each environment's centre, geometric-mean cost and mean score.

Figure 5. Quality against expense: each deck's total score (out of 20) and its cost, in quadrants split at the median cost and the median score.

Model	Mode	Alignment/5	Art theme/5	Crop/5	Layout/5	Total/20
Opus 5.5	Bespoke	4 3 · 5	3.5 4 · 3	2 1 · 3	4.5 4 · 5	14 12 · 16
	Computer use	5	4	3	5	17
	Bowerjack	5 5 · 5	4.5 5 · 4	2 4 · 0	4 3 · 5	15.5 17 · 14
	All modes	**4.6**	**4**	**2.2**	**4.4**	**15.2**
Sonnet 5.5	Bespoke	4 5 · 3	2.5 1 · 4	2 2 · 2	5 5 · 5	13.5 13 · 14
	Computer use	4	3	3	4	14
	Bowerjack	5 5 · 5	3 4 · 2	3 4 · 2	4.5 5 · 4	15.5 18 · 13
	All modes	**4.4**	**2.8**	**2.6**	**4.6**	**14.4**
Astra 6	Bespoke	4.5 5 · 4	5 5 · 5	4.5 4 · 5	2.5 3 · 2	16.5 17 · 16
	Computer use	4	5	5	1	15
	Bowerjack	5 5 · 5	1 0 · 2	3 3 · 3	3 4 · 2	12 12 · 12
	All modes	**4.6**	**3.4**	**4**	**2.4**	**14.4**
Sol 6.1	Bespoke	3.5 3 · 4	4.5 5 · 4	4.5 5 · 4	3 3 · 3	15.5 16 · 15
	Computer use	4	4	0	2	10
	Bowerjack	5 5 · 5	4 3 · 5	5 5 · 5	2.5 3 · 2	16.5 16 · 17
	All modes	**4.2**	**4.2**	**3.8**	**2.6**	**14.8**

Mean score of each model's decks in each mode (the trial scores beneath, where there are two), and across all modes. Darker is better. Computer use was run once per model (Sol 6.1's run is the October 16 pilot), and Astra 6's computer-use deck has 7 of 12 cards.

Figure 6. Score on each criterion by model and environment: the mean of each model's decks, with each trial's score beneath, and across all modes.

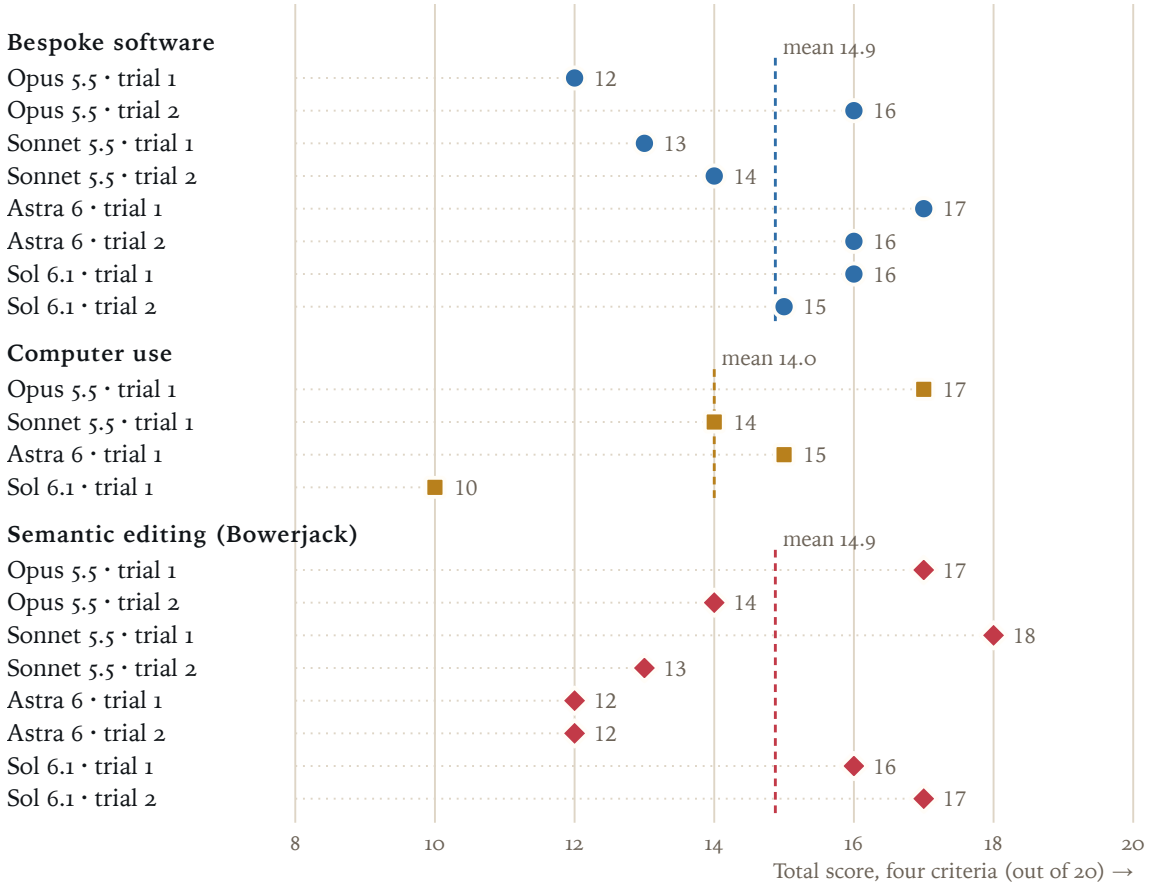
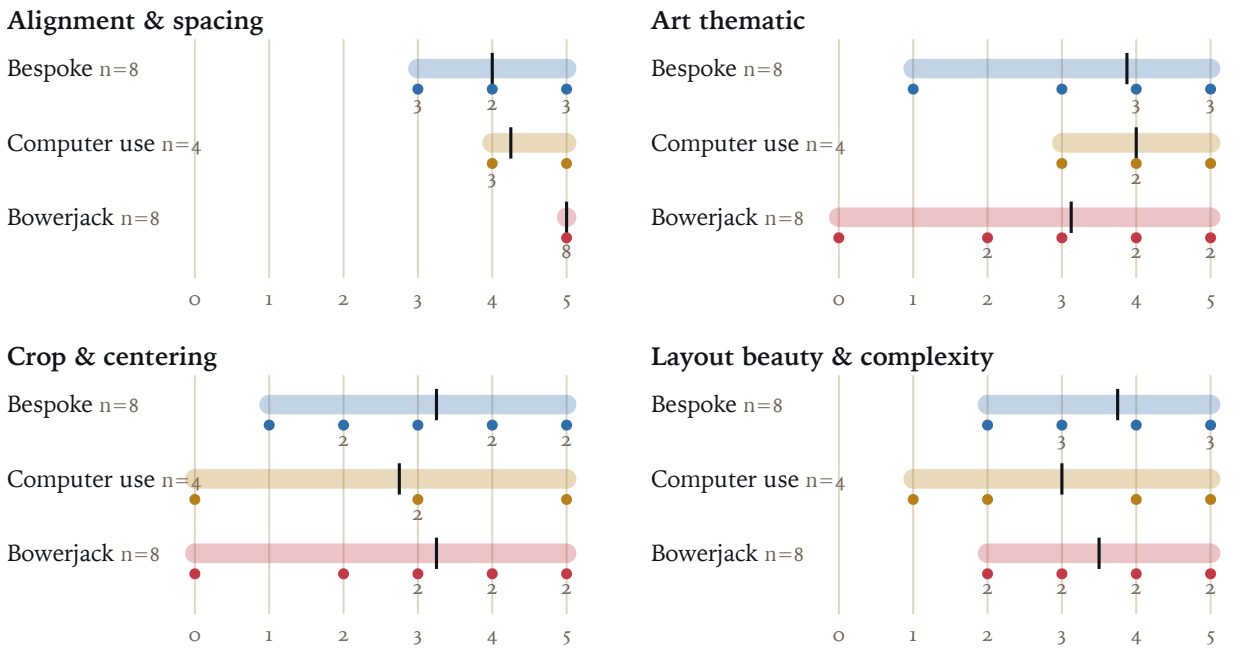


Figure 7. Total quality score per deck (out of 20), grouped by environment; the dashed line marks each environment's mean.



Band: lowest to highest score · black tick: mean · dot below: a score given, with how many decks gave it when more than one.

Figure 8. Score range for each criterion by environment: the lowest to the highest score, the mean, and every deck.

6.2 Correctness and rights review

Every deck was also checked for what could put a commercial printer in legal trouble and for what prints wrong. Each picture's license and credit were checked against its source record (Wikimedia Commons, the Metropolitan Museum of Art or the Cleveland Museum of Art). A rights violation is a license that requires attribution with the credit wrong or missing, or a recorded license that is wrong or missing. A printed error is something on the card that is wrong: a picture whose source names another god, or a page of the wrong size. A printed doubt is something on the card that may be wrong: an identification its source disputes, or a frame within ⅛ in of the cut, where cutting drift shows. The roster and rank choices are left to the agents, and nothing that does not print is counted.

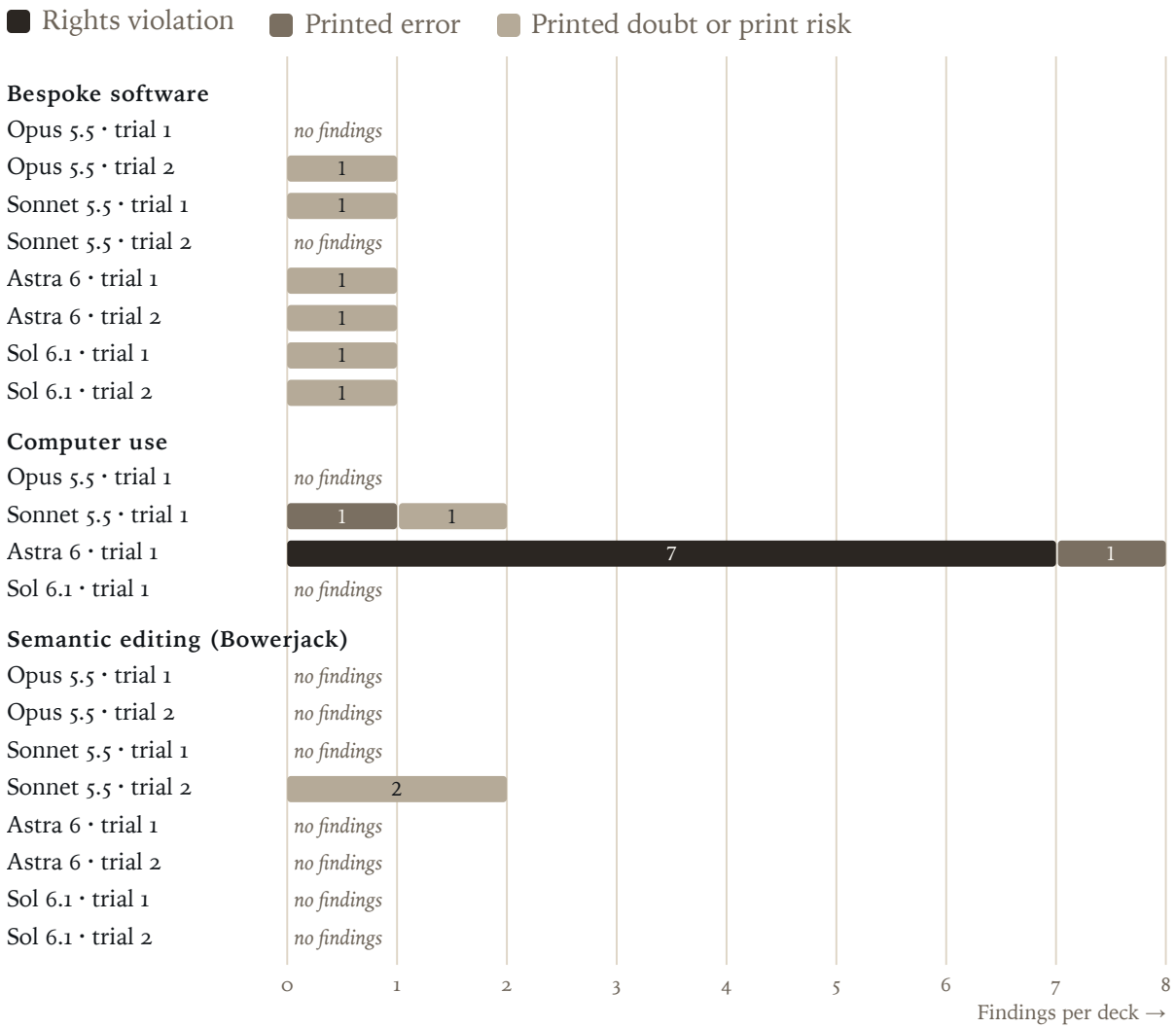


Figure 9. Correctness and rights findings per deck, by severity.

SECTION 7

Conclusion

Across all decks, the most common weakness was cropping and centering, followed by thematic consistency of the art. Layout quality varied by model: Claude models produced more attractive layouts than GPT models in every environment, while GPT models produced better crops except in Sol's computer-use pilot. Every Bowerjack deck was perfectly aligned, and alignment errors appeared in every other environment. Although AI research makes gathering creative material fast, a human editor remains best placed to choose among art options, set a theme and judge a crop.

Computer use is best reserved as a last resort, given its cost, slow runs and print errors. Of the remaining approaches, semantic document editing used fewer tokens and finished faster than bespoke software at the median, validates picture rights as it works, and produces a document that

people and agents can both edit. Until agents match the taste of skilled designers, the most effective way to use AI for desktop publishing is semantic document editing, with agents doing the research and layout and people refining the result.

Works Cited

Awasthi, Palash. "500,000 Tokens to Click a Dropdown." *Reflex*, 27 Apr. 2026, https://reflex.dev/blog/vision-agents-vs-api-calls/.

"Cleveland Museum of Art Open Access." *The Cleveland Museum of Art*, www.clevelandart.org/open-access. Accessed 18 Oct. 2026.

"The Met Collection API." *The Metropolitan Museum of Art*, metmuseum.github.io. Accessed 18 Oct. 2026.

Wikimedia Commons. Wikimedia Foundation, commons.wikimedia.org. Accessed 18 Oct. 2026.

APPENDIX A

Trial Decks

Contact sheets of every deck the trials produced, each card trimmed at its cut line.

A.1 Bespoke software

Claude Opus 5.5 · trial 1



Claude Opus 5.5 · trial 2



Claude Sonnet 5.5 · trial 1



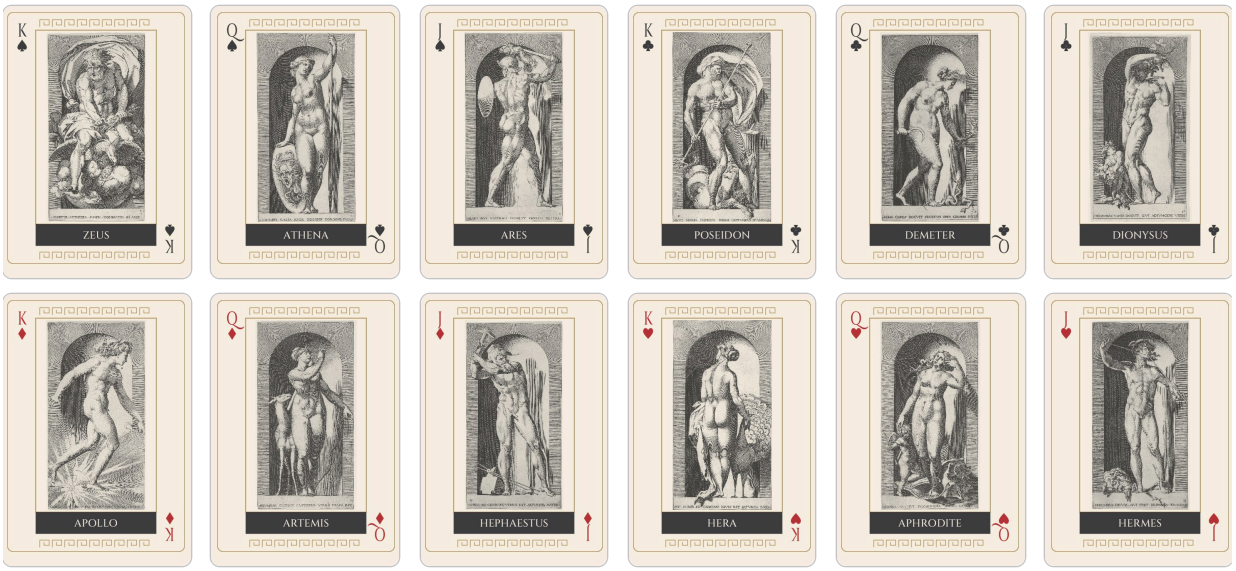
Claude Sonnet 5.5 · trial 2



GPT Astra 6 · trial 1



GPT Astra 6 · trial 2



GPT Sol 6.1 · trial 1



GPT Sol 6.1 · trial 2



A.2 Computer use

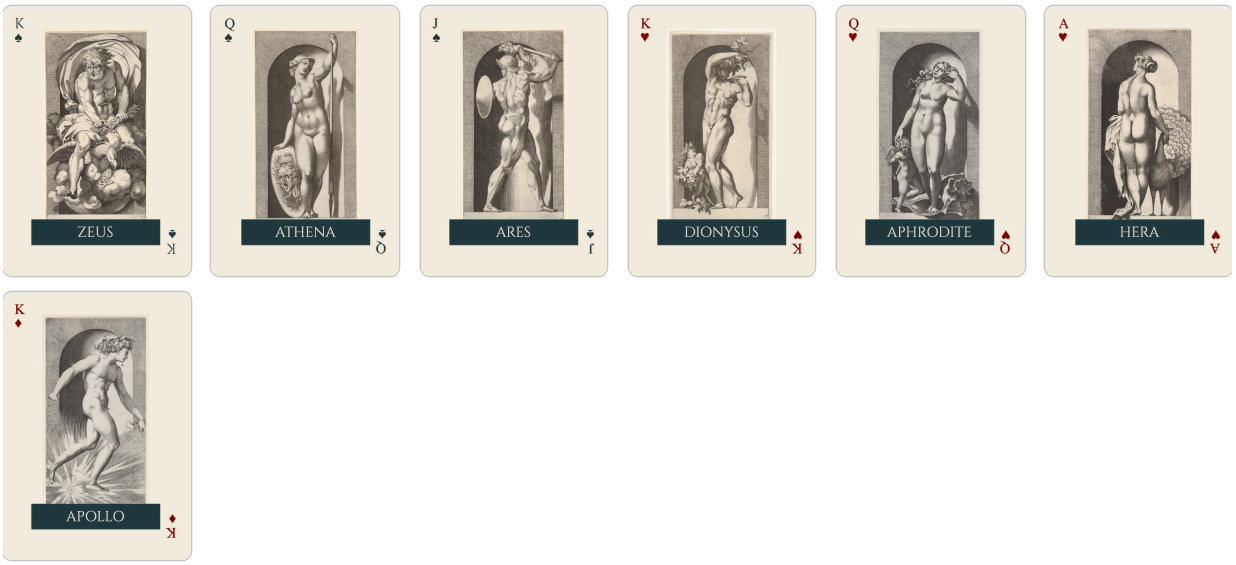
Claude Opus 5.5 · trial 1



Claude Sonnet 5.5 · trial 1



GPT Astra 6 · trial 1 · Partial deck: 7 of 12 cards.



GPT Sol 6.1 · trial 1



A.3 Semantic editing (Bowerjack)

Claude Opus 5.5 · trial 1



Claude Opus 5.5 · trial 2



Claude Sonnet 5.5 · trial 1



Claude Sonnet 5.5 · trial 2



GPT Astra 6 · trial 1



GPT Astra 6 · trial 2



GPT Sol 6.1 · trial 1



GPT Sol 6.1 · trial 2



APPENDIX B

Bowerjack in the Agents

The semantic document-editing trials as they ran: the same Bowerjack Studio, driven by ChatGPT through WebMCP and by Claude through its connector.

B.1 Bowerjack inside of ChatGPT

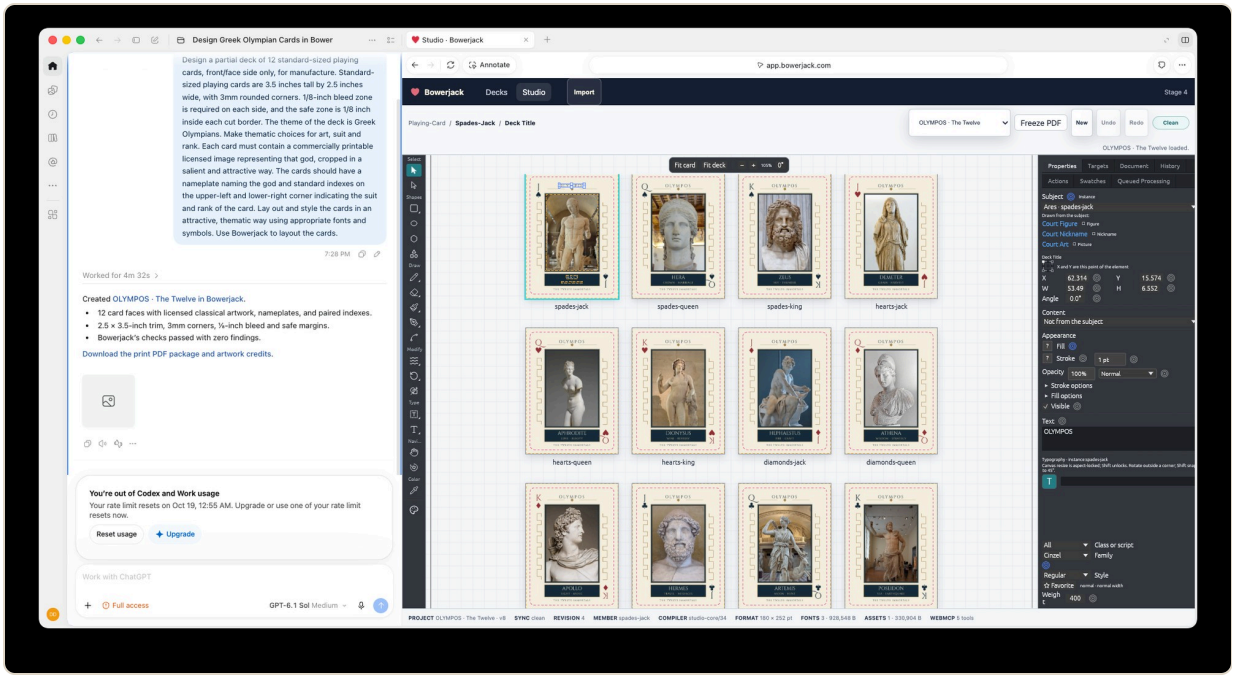


Figure 10. Bowerjack Studio open in ChatGPT's browser beside the Sol 6.1 conversation that built the deck.

B.2 **Bowerjack connector in Claude**

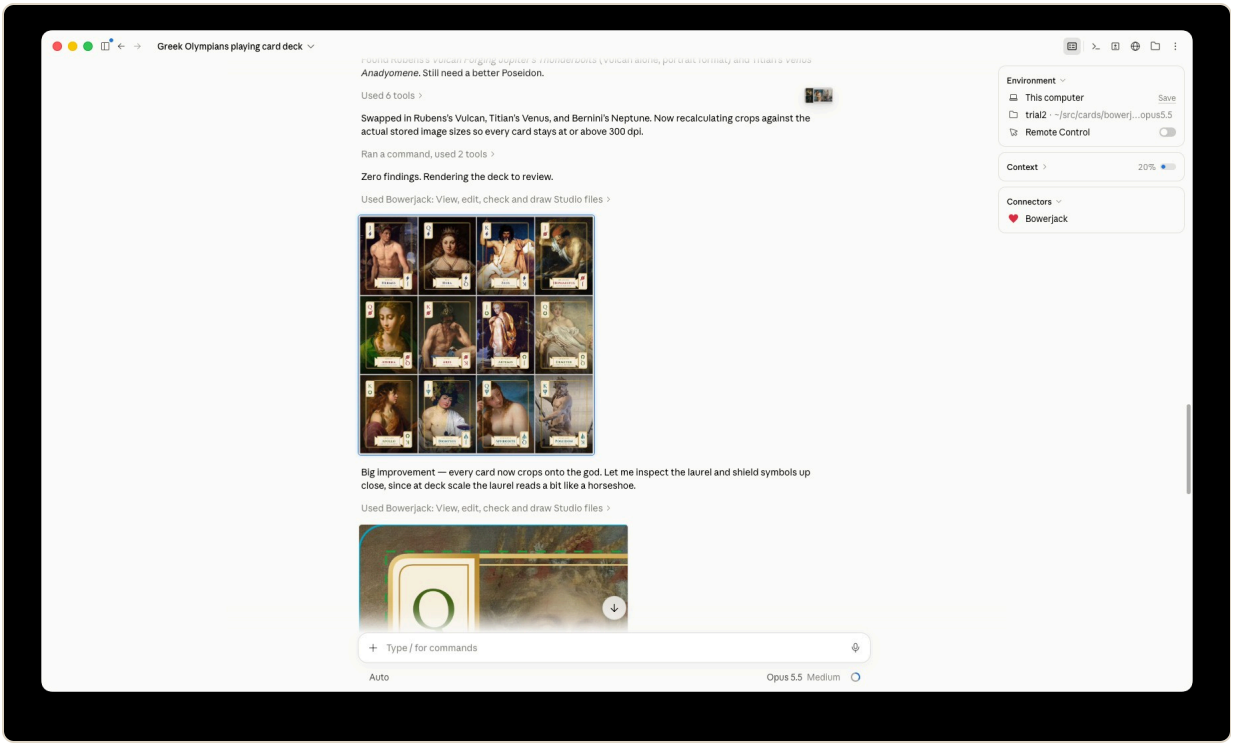


Figure 11. The Bowerjack connector in a Claude session, with a contact sheet Claude rendered to review the deck.

B.3 Bowerjack session being controlled by Claude

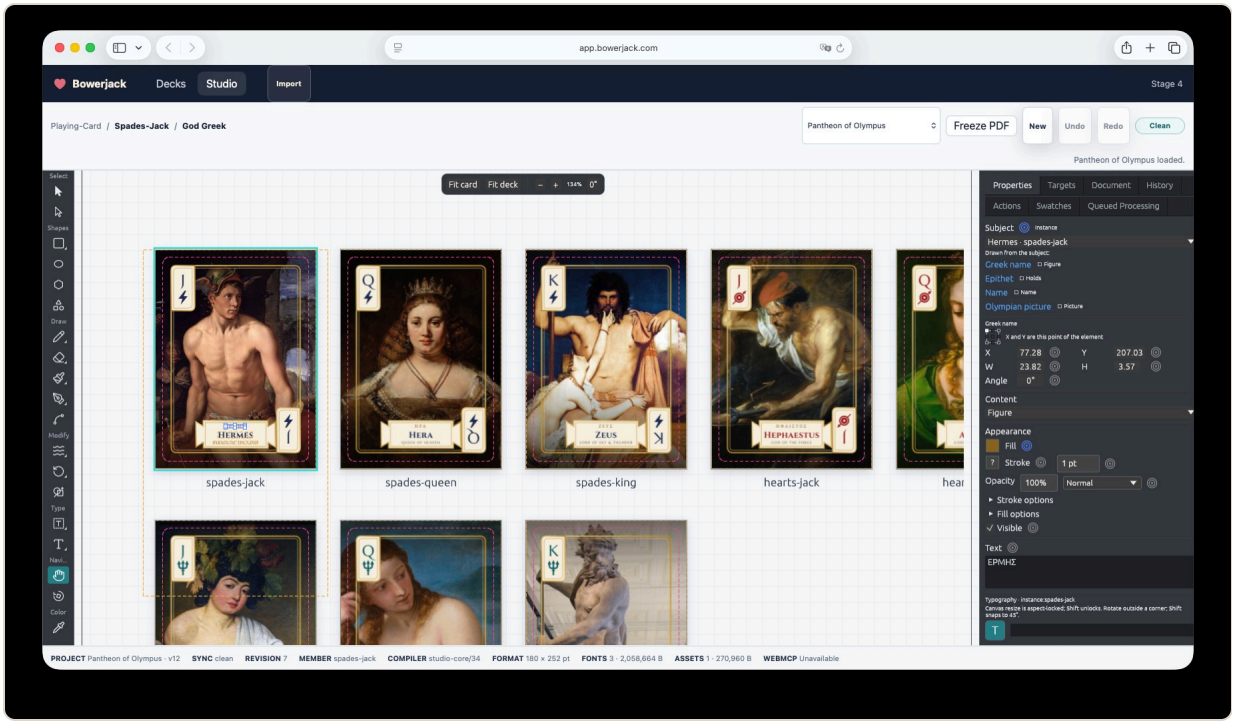


Figure 12. The same Bowerjack Studio project in the browser while Claude edits it; changes appear live.