

BUSINESS DEVELOPMENT

Case Pitch Builder

Research a newly filed case and turn it into a client-ready pitch deck or memo, in any library.

HOW TO USE THIS DOCUMENT

Upload this PDF to the AI assistant you use with the Docket Navigator connector and ask it to create a skill from it. The pages that follow are the complete skill: the numbered sections are its instructions, and its frontmatter is listed at the end. The assistant needs the Docket Navigator MCP connector connected and authenticated before the skill will run.

Turns a new filing into a pitch. The attorney learns a case was filed, and needs enough verified fact by a deadline to walk into a room and be the firm that already did the work.

The deliverable is client-facing. That governs everything below. A figure that cannot be traced to a Docket Navigator object does not go in it.

The five stages

STAGE	QUESTION	OUTPUT
1. Anchor	What was filed, and what else moved with it?	Case ids, forums, dates
2. Foundation	Six research threads	Verified narrative
3. Substantiate	Show me the source	Resolvable links
4. Build	The artifact	Deck or memo
5. Refine	Deepen, revise, list what to read	Revised artifact

Do not skip stage 3. Facts degrade when they move from conversation into slides, and the check has to happen before the move.

Stage 1: Anchor the filing

Search the case target for the party and date window. Read the case profile for every hit.

The profile returns `eligible_tabs`. **That list is the map for this library.** It names every target available for this case, with its index. Do not assume a tab exists in one library because it exists in another, and do not assume one is absent without reading the list.

Filings that land within a few days of each other across different forums are usually one coordinated action. Group them. Set apart anything that only shares a party name.

Stage 2: Build the foundation

Six threads. Run them in this order, because each narrows the next.

T1. The filing.

Forum, number, judge, date, asserted rights, counsel, current docket state. Read the docket entries, not just the header. Service dates and response deadlines live there and are usually the reason the pitch is urgent.

T2. The party pair.

How much history do these two have, and across which forums? See the AND/OR note under Search construction before running this.

T3. Each asserted right, separately.

Prior proceedings, prior determinations, prior constructions, prior agency review. **Run this per right, never as a set.** Two rights asserted in one complaint routinely have opposite records, and the difference is usually the most useful thing you will find.

T4. The decision that reorders strategy.

From T3, identify any prior adverse or favorable determination that a party facing this right now would have to plan around, and check whether it was appealed and how it came out. This is the single highest-value fact in most pitches.

T5. The assigned judge.

Caseload, determinations, motion outcomes, time to milestones, motion pendency. Read the motion orders themselves rather than only the success rates. Judges often state their own practice in the order text, and a stated practice is worth more to a client than a percentage.

T6. Parallel proceedings.

How the new case sits against the companion actions. Shared rights across forums create stay and estoppel questions that are usually the client's real concern.

Close with a short list of the facts that would actually move a general counsel, each with its source. That list is what gets pitched. The rest is support.

Stage 3: Substantiate

For every fact flagged as decision-moving, return a resolvable link and quote the coded value verbatim.

Resolve document references into full links. Read the coded fields as they are written rather than paraphrasing them into ordinary language, because the paraphrase is usually where the error enters.

State the limits of what the coded field establishes. A determination recorded against a registered right is recorded at the level of that right, so it may not tell you which individual claims, marks, or counts were resolved. If the answer matters, say the underlying document has to be read.

Stage 4: Build the artifact

Establish a visible convention that separates **record** from **analysis**, and apply it to every statement in the deliverable. Two labels, used consistently, beat any amount of hedging prose.

Rules for the artifact:

- Every slide or section stating a figure names the Docket Navigator object behind it.
- No figure appears that did not appear in the conversation first.
- Any aggregate you counted yourself is labeled as derived. Do not present a counted figure as a Docket Navigator statistic.

- No prediction is attributed to Docket Navigator.
- Put the unverified items on an early page, in the open, before the client finds them.

Close by naming what still needs checking and what would change if it comes back the other way.

Stage 5: Refine

Expect three follow-ups: deepen one thread, produce a short reading list, and revise the artifact. On revision, modify what exists rather than regenerating, and check the source lines survived. That is where sourcing is most often lost.

Search construction

Work in this order every time. It is faster than guessing and it is the only reliable path across libraries.

1. `docnav_plan_search_workflow` with the library and the question in plain language. It returns a ranked recipe with the correct target, date basis, and denominator.
2. `docnav_lookup_search_values` for any named entity or taxonomy value. Ids are namespace-scoped.
3. Rerun the planner with resolved entities so it composes the criteria itself.
4. `docnav_validate_and_preview_search` before relying on anything.

Two shortcuts worth knowing:

Profile binder tabs carry correct saved searches.

Inspecting a tab returns a working search handle for that library and target. Copying it is more reliable than assembling criteria by hand, and it is the fastest way to learn a criterion's required value shape in a library you have not worked in.

Criteria compose in two directions.

Separate criteria AND together. Values inside one criterion OR together. To find matters involving two parties, use two criteria with one name each. Putting both names in one criterion returns everything involving either of them, which on an active pair can be many times larger and looks entirely reasonable on a slide. Read the returned description to confirm the shape before using the count.

Use docnav_discover_search_options when the planner has no recipe that fits. It is the fallback, not the starting point.

Object discipline

These distinctions are what separate a credible pitch from an embarrassing one. They hold across libraries.

KEEP APART	BECAUSE
A decision to review from a decision on the merits	Granting review says nothing about the outcome
An agency determination from a court holding	Different standards, different burdens, different binding effect
A summary affirmance from a reasoned opinion	A summary affirmance has no reasoning to distinguish and sets no precedent
A recorded outcome from a party's argument	Docket Navigator records what was decided, not what was contended
Coded roles from caption position	The declaratory plaintiff is often the challenger. Read the coded role
A right-level determination from a claim-level one	Determinations are recorded per registered right
A counted aggregate from a product statistic	If you counted rows, say you counted rows
An original outcome from a post-appeal one	Displayed outcomes are original results unless appeal data says otherwise

One further rule. Never infer the content of an argument from the fact that it failed. A losing petition tells you the outcome. It does not tell you what was argued.

What Docket Navigator does not hold

Say so plainly when you hit one of these. Stating a limit is a credibility asset in a pitch, and guessing past it is the fastest way to lose the client.

Docket Navigator records proceedings, coded events, decisions, and the documents that carry them. It does not hold the text of the underlying registered rights, the substance of party briefing, or anything that has not yet been docketed and coded.

Before saying a thing is absent, check that you looked in the right place. Open the relevant profile and read `eligible_tabs`. A zero from a tab that exists is a finding. Silence from a tab you never opened is not.

When the answer requires something outside the record, name the source that would settle it and say what would change if it came back either way.

Common failure modes

Running the asserted rights as a set.

Hides the asymmetry that is usually the whole story.

Reporting a party-pair count from a single criterion.

Wrong by a large multiple, and it looks normal.

Quoting a rate without reading the orders.

The stated practice in the order text is worth more than the percentage.

Letting a source line drop during revision.

Check slide by slide after any rework.

Filling an empty record with plausible history.

A right with no prior proceedings has no prior proceedings. That is a finding, and it is often the most valuable one in the deck.

Treating the docket as static.

Counsel appears, orders issue, counts climb. Note the date your facts were pulled, and flag anything time-sensitive as perishable.

Skill frontmatter

These fields are the skill's YAML frontmatter. Everything before this section is the skill body.

name	case-pitch-builder
description	Research a newly filed case in Docket Navigator and turn it into a client-ready pitch deck or memo, in any library. Use when an attorney wants to pitch a party on representation in a new matter, needs the factual foundation on a case, a party pair, an asserted right, or an assigned judge, or asks what a defendant facing this now would need to know. Triggers on "pitch", "new filing", "who should we call", "build the foundation on", "what's the history between", "what has this judge done with", or any request for a deck or memo about a case. Works across the patent, trademark, copyright, trade secret, antitrust, securities, SEC, UPC, and omnibus libraries. Prefer this skill over ad hoc searching whenever the output will be shown to a client.