FACILITATOR'S GUIDE

MCP Integration Lab Panel Delivery & Presenter Notes

45-minute panel session prep for Technical Enablement Manager interview

Prepared by John Yzaguirre
Format 45-minute panel presentation + Q&A
Companion to MCP Integration Lab Design Document

SESSION OVERVIEW

Panel session structure

How to deliver the MCP Integration Lab design in a 45-minute evaluative panel.

Three-part format

The session is designed to demonstrate three competencies: technical communication, enablement design thinking, and improvisation under pressure.

Part 1: Presentation (8 min)

Structured walk-through of the lab design. Lead with the customer story, not the technology. Show you can distill a complex technical topic into a clear enablement narrative.

Part 2: Deep walk-through (20 min)

Section-by-section lab walk-through with the panel. Demonstrate depth of knowledge, design rationale, and ability to adapt to questions mid-flow. This is where you prove you can teach.

Part 3: Q&A + Discussion (17 min)

Open floor for panel questions. Show you can think on your feet, handle tough questions honestly, and bridge back to the core narrative. Finish with a strong close.

Evaluator signals to hit

What the panel is likely assessing:

- **Technical credibility:** You understand MCP, RAG, and enterprise integration patterns at depth
- **Enablement instinct:** You design for the learner, not the technology. Labs teach skills, not just concepts
- **Customer empathy:** You connect technical decisions to customer outcomes and SE conversations
- **Practical judgment:** You make smart trade-offs (realism vs. time-box) and can explain why
- **Communication clarity:** You can explain complex topics simply without dumbing them down
- **Poise under pressure:** You handle unexpected questions, admit what you don't know, and redirect constructively

Success metric

If the panel walks away thinking "I'd trust this person to run our technical enablement program," you've succeeded. Every talking point should reinforce that trust.

What great looks like

Criteria for a top-performing delivery, mapped to likely evaluation dimensions.

Dimension	Good	Great
Opening	States what the lab is and what it covers	Starts with a customer pain point that makes the lab feel inevitable and urgent
Technical depth	Explains MCP architecture correctly	Shows trade-off reasoning, anticipates production questions, connects to competitive landscape
Enablement design	Describes lab activities clearly	Explains why each activity exists, what skill it builds, and how you'd measure success
Customer lens	Mentions customer relevance	Every design decision ties back to an SE customer conversation or demo moment
Q&A handling	Answers questions accurately	Reframes questions constructively, admits gaps honestly, bridges back to the lab's value
Closing	Summarizes the lab	Leaves the panel with a memorable statement about what this lab enables for the SE team

Three themes to weave throughout

THEME 1: "HANDS-ON, NOT SLIDES"

SEs learn by doing. This lab puts them in the driver's seat building real infrastructure, not watching a demo. Every section ends with a "done looks like" checkpoint.

THEME 2: "ENTERPRISE SEARCH, NOT TOY DEMOS"

The lab teaches the exact integration pattern customers ask about: enterprise search with grounding, citations, and permissions. This is how Glean positions workplace AI.

THEME 3: "FROM LAB TO CUSTOMER CONVERSATION"

Every technical activity connects to a customer-facing talk track. SEs don't just learn how MCP works; they leave with the vocabulary and confidence to position it in their next meeting. The lab is a bridge from technical understanding to sales impact.

Delivery mindset

You're not presenting a document. You're showing how you think about enablement. The panel wants to see your design process, your prioritization instincts, and your ability to make complex things clear. Let the lab speak for itself; your job is to illuminate the reasoning behind it.

MINUTE-BY-MINUTE

Detailed session timing

Segment-by-segment delivery plan with talking points and visual cues.

Part 1: Presentation (0:00 – 8:00)

Structured overview. You control the flow. Keep it crisp and narrative-driven.

0:00-1:30

Opening hook

Start with the customer problem, not the lab title. "Imagine your largest enterprise customer asks: 'How do we give our AI agents controlled access to our internal systems without building a custom integration for every one?' That's the question this lab teaches SEs to answer."

1:30-3:30

Lab overview and architecture

Describe the four-box architecture (Agent → MCP Client → MCP Server → Enterprise API). Explain why enterprise search is the right use case. Reference the three-layer pattern SEs will build and learn to explain.

3:30-5:30

Seven-section flow

Quick pass through the lab sections. Don't go deep yet (that's Part 2). Hit the highlights: customer story setup, infrastructure, mock API, MCP wiring, validation, troubleshooting, positioning. Emphasize the "done looks like" checkpoints.

5:30-7:00

Key design decisions

Highlight 2-3 of the most interesting trade-offs: (1) Why stdio over Streamable HTTP, (2) Why mock data over live search, (3) Why one tool instead of many. Show you think about realism vs. time-box deliberately.

7:00-8:00

Bridge to walk-through

"That's the overview. Let me take you through each section in detail so you can see how the design decisions play out in practice. I'm happy to go deeper on anything as we go." This invites the panel to engage interactively.

Timing trigger

If you're past 8:00 and still in Part 1, skip the bridge statement and transition directly: "Let me jump into the section details." Don't compress Part 2 to make up time; it's the most important segment.

MINUTE-BY-MINUTE (CONTINUED)

Walk-through and Q&A timing

Part 2: Deep walk-through (8:00 – 28:00)

Section-by-section with the panel. Adapt pace based on their engagement and questions.

8:00-11:00

Sections 1-2: Customer story + environment setup

Emphasize why the customer narrative comes first. Explain how the sketch exercise builds the mental model. For setup, stress that CloudShell eliminates local toolchain issues. **Key point:** "The lab starts with the why, not the how."

11:00-16:00

Sections 3-4: Mock API + MCP server wiring

Walk through the `/search` contract and the `kb_search` tool code. Highlight metadata fields (title, snippet, score, acl_group) and explain why each exists. Discuss the stdout/stderr rule. **Key point:** "This is the section where SEs go from understanding MCP conceptually to having built something with it."

16:00-21:00

Sections 5-6: Validation + troubleshooting

Discuss the three validation dimensions (relevance, grounding, permissions). Walk through the troubleshooting table and the "client, server, endpoint" triage pattern. **Key point:** "Troubleshooting during a customer POV is where SE credibility is won or lost. This section builds that muscle."

21:00-25:00

Section 7 + success criteria + reasoning

Cover the positioning narrative exercise and the "production delta" conversation. Explain success criteria and the assessment approach. Touch on trade-offs table and enablement fit. **Key point:** "We measure success by whether SEs can demo MCP to a customer tomorrow, not by quiz scores."

25:00-28:00

Stretch ideas + evolution roadmap

Brief pass through v2/v3/v4 evolution and stretch ideas. Show the lab is a platform, not a one-off. **Key point:** "The lab structure is a reusable template. Swap the mock search for any enterprise system to create a library of integration labs."

Part 3: Q&A + Close (28:00 – 45:00)

28:00-40:00

Open Q&A

Let the panel drive. Keep answers concise (60-90 seconds max). Bridge back to the lab when possible. See the Anticipated Q&A pages for prep.

40:00-45:00

Strong close

Prepared closing statement. "One protocol, many enterprise systems, grounded answers. This lab gives SEs the hands-on confidence to demo, position, and troubleshoot MCP integrations with real customer impact."

TALKING POINTS

Key narratives to deploy

Scripted phrases for high-impact moments. Use when the moment calls for precision.

Opening hook

> "Imagine your largest enterprise customer asks: 'We have fifty internal systems our AI agents need to reach. Do we build fifty custom integrations, or is there a better way?' That's the question this lab teaches SEs to answer with confidence."

Why enterprise search

> "I chose enterprise search as the use case because it's the most common integration conversation SEs have. Retrieve authoritative context first, generate grounded answers second. That's the RAG pattern, and it's the core of how Glean positions workplace AI to customers."

Design philosophy

> "The lab is designed around a simple principle: SEs learn by doing. Every section ends with a 'done looks like' checkpoint. It's not about watching a demo; it's about building something, breaking it, fixing it, and then explaining it to a customer."

Trade-off reasoning

> "Every simplification in the lab is deliberate. Stdio instead of Streamable HTTP saves twenty minutes of OAuth setup. Mock data instead of live search makes debugging reproducible. One tool instead of many keeps the concept set small. Each of these is explicitly called out so SEs know what the production delta looks like."

Production delta

> "The protocol is production-ready. For a customer's environment, we'd add OAuth-aligned authorization, Origin header validation for HTTP transports, and least-privilege tool scoping. The lab teaches SEs to have that conversation confidently."

Reusable template

> "The lab structure is a reusable template. Mock API, MCP server, validation, troubleshooting, positioning. Swap the mock search for a ticketing system, CRM, or document store and you have a library of integration labs from the same foundation. One architecture, many customer stories."

Strong close

> "One protocol. Many enterprise systems. Grounded answers. This lab gives SEs the hands-on confidence to demo, position, and troubleshoot MCP integrations with real customer impact. That's the outcome I'd measure, and that's what I'd build the enablement program around."

TALKING POINTS (CONTINUED)

Situational phrases

Context-specific phrases for navigating common panel moments.

When asked "Why not use a real Glean endpoint?"

> "Great question. A mock endpoint gives us three things: reproducible results for validation, deterministic data for debugging exercises, and zero dependency on production systems during a training lab. In a follow-up session, SEs could connect to a sandbox Glean instance as a stretch activity."

When asked about scaling to a larger SE team

> "The lab is designed to scale in two directions. First, it works as an instructor-led session for cohorts of 8-15 SEs. Second, we can package it as a self-service GitHub repo with Terraform templates so new SEs run through it independently during onboarding. That's the v4 stretch goal."

When asked about measuring effectiveness

> "I'd measure three things. First, lab completion: can the SE run an end-to-end query and get a grounded answer? Second, demo confidence: a pulse check asking 'How ready are you to demo MCP tomorrow?' Target is 4+ out of 5. Third, deal impact: track whether SEs who complete the lab reference MCP in customer conversations at a higher rate."

When asked about competitive differentiation

> "MCP is an open standard with broad ecosystem adoption. The competitive advantage isn't the protocol itself; it's that our SEs can explain how enterprise search connects to agent workflows with a real integration pattern. That's the story competitors struggle to tell because they're either model-first or connector-first, not both."

When you don't know the answer

> "That's a great question, and I want to be honest that I don't have a fully formed answer on that specific point. Here's how I'd approach it: [describe your process]. I'd want to research that before committing to an approach. What I can tell you is [bridge to something you do know confidently]."

When redirecting to the lab's value

> "That's an important consideration. The way it connects back to the lab is [specific connection]. The core principle is that SEs need hands-on experience with the integration pattern, not just a conceptual overview, because that's what gives them credibility in customer conversations."

> **Delivery tip**
>
> These phrases are starting points, not scripts. Internalize the key ideas and deliver them naturally. The panel will notice if you're reading verbatim. Aim for the confidence of someone who has taught this material before.

ANTICIPATED Q&A

Likely questions and strong answers

Structured prep for the Q&A segment. Each answer bridges back to the lab's value.

Category	Likely question	Strong answer
DESIGN	Why 60-90 minutes and not a half-day?	Scope is one tool, one integration pattern. Longer sessions add fatigue without adding conceptual depth. v2 (Streamable HTTP + auth) naturally extends to half-day when we're ready.
DESIGN	How do you handle different skill levels in the room?	The "done looks like" checkpoints create natural catch-up points. Advanced SEs can move to troubleshooting or stretch ideas while others finish core activities. Pair work mixes skill levels naturally.
TECHNICAL	Why stdio instead of Streamable HTTP?	Stdio eliminates OAuth, Origin validation, and HTTP server setup. That's 20+ minutes of cognitive overhead that crowds out core concepts. We teach Streamable HTTP conceptually and implement it in v2.
TECHNICAL	Is MCP actually production-ready?	The specification is stable and well-documented. The "production delta" is auth and transport hardening, which is standard for any protocol. The lab explicitly teaches SEs to have this conversation.
CUSTOMER	What customer objections do SEs face about MCP?	"Is it secure?" (yes, with proper auth and scoping), "Is it vendor-locked?" (no, it's an open standard), "Does it work with our systems?" (MCP servers exist for many enterprise categories). The lab prepares SEs for all three.
CUSTOMER	How does this help SEs close deals?	SEs who can demo a real integration pattern, explain grounding, and troubleshoot in real time build more credibility than those showing slides. Technical credibility accelerates deal velocity.
SECURITY	Isn't auth: NONE on the Lambda a security risk?	It's a lab simplification, explicitly called out. The Lambda exists for 60 minutes in a personal account and is deleted after. Production requires IAM or OAuth. Teaching this distinction is part of the lab.

ANTICIPATED Q&A (CONTINUED)

More likely questions

Category	Likely question	Strong answer
SECURITY	What about prompt injection through the MCP tool?	The mock data is deterministic, so injection risk is minimal in the lab. For production, we teach tool scoping, input validation, and human-in-the-loop controls. It's a stretch module idea for v2.
SCALE	How do you roll this out globally?	Three phases: (1) pilot with one region's SEs, iterate based on feedback, (2) train-the-trainer for regional leads, (3) self-service kit for async completion. Each phase builds on the previous.
SCALE	What if SEs get stuck during the lab?	The troubleshooting section explicitly teaches the debugging framework. Additionally, the cheat sheet provides quick-reference commands and common failure patterns. Pair work means no one is stuck alone.
DESIGN	Why Python and not TypeScript?	The MCP Python SDK (FastMCP) has the cleanest decorator-based API and matches the most popular MCP tutorials. Python is also more readable for SEs with mixed programming backgrounds. The concepts transfer to any SDK.
TECHNICAL	How does this relate to Glean's actual API?	The mock `/search` contract mirrors a simplified version of real enterprise search: query in, ranked results with metadata out. When SEs connect to a real Glean endpoint, the integration pattern is identical; only the URL and auth change.
CUSTOMER	What if a customer already uses a competitor's connector framework?	MCP is an open standard, not a vendor lock-in play. The pitch is: "You can keep what works and add MCP for new agent integrations." Standards reduce switching cost for the customer, which is a strength, not a weakness.
DESIGN	How do you keep the lab current as MCP evolves?	The lab is built on the stable core of MCP (tools, schemas, transports). Protocol evolution adds new capabilities (like Streamable HTTP replacing SSE) that we fold into version updates. The structure is durable; the details are versioned.

Q&A strategy

Keep answers to 60-90 seconds. If a question goes deep, offer to go deeper: "I can dive further into that if it's useful, or I can address other questions first." This shows respect for the panel's time and gives you control of pacing.

DIFFERENTIATORS

What makes this lab stand out

Five design choices that differentiate this from a typical enablement exercise.

1. ENTERPRISE SEARCH, NOT A TOY DEMO

Most MCP tutorials use trivial examples (weather APIs, file readers). This lab uses enterprise search with ranked results, metadata, and permissions because that's the integration pattern SEs actually encounter in customer conversations. The use case is the teaching tool.

2. HANDS-ON BUILDING, NOT SLIDE-WATCHING

SEs deploy infrastructure, write code, and validate results. Every section has a "done looks like" checkpoint so both the learner and facilitator know when the skill is acquired. The lab measures doing, not just understanding.

3. RAG GROUNDING AS A FIRST-CLASS CONCEPT

The lab explicitly teaches the retrieve-then-generate pattern with three validation dimensions: relevance, grounding, and permissions. SEs leave understanding not just how MCP works, but why grounded answers are the enterprise value proposition.

4. TROUBLESHOOTING AS A SKILL, NOT AN AFTERTHOUGHT

A dedicated section on fault isolation teaches the "client, server, endpoint" triage pattern. SEs deliberately break each layer and fix it. This builds the confidence needed for live demos and customer POVs, where things will inevitably go wrong.

5. CUSTOMER POSITIONING BUILT INTO THE LAB

The final section converts technical experience into customer-ready talk tracks. SEs practice a two-minute positioning narrative and a one-minute architecture explanation with a partner. The lab doesn't end with "you understand MCP"; it ends with "you can sell MCP."

When to use this page

If the panel asks "What's different about your approach?" or "Why should we use this instead of existing MCP tutorials?", these five points are your answer. Lead with whichever point feels most relevant to the question.

DIFFERENTIATORS (CONTINUED)

Why evaluators should notice

Connecting the lab design to what the role requires.

Mapping lab design to role competencies

Role competency	How the lab demonstrates it
Technical depth	Lab covers MCP architecture, JSON-RPC transport, tool schemas, RAG grounding, and production considerations. Code samples are real and runnable.
Enablement design	Clear learning objectives, "done looks like" checkpoints, mixed-skill accommodations, and measurable success criteria. Lab follows adult learning principles.
Customer orientation	Every design decision maps to a customer conversation or SE demo moment. Talk tracks are built into the lab, not added as an afterthought.
GTM awareness	Enterprise search use case aligns with how the company positions workplace AI. Lab teaches the exact narrative SEs need for customer conversations.
Scalability thinking	Reusable template design, self-service packaging roadmap, and train-the-trainer model show how to scale beyond a single session.
Practical judgment	Deliberate simplifications (stdio, mock data, single tool) are explicitly documented with production alternatives. Shows mature trade-off reasoning.

The "would I hire this person?" test

The panel is ultimately asking themselves one question. Here's how the lab answers it:

Can they teach?
The lab breaks a complex protocol into seven progressive sections, each building on the previous. The "done looks like" pattern shows pedagogical rigor.

Can they think technically?
The architecture diagram, code samples, and trade-off analysis show genuine technical understanding, not surface-level familiarity.

Can they connect to business outcomes?
Talk tracks, positioning narratives, and the "demo-to-POV bridge" concept show a direct line from lab activity to revenue impact.

Can they build a program?
The reusable template design, iteration roadmap, and scalability thinking show someone who builds systems, not just one-off sessions.

FACILITATION

Facilitation tips for the panel

How to deliver with confidence, handle curveballs, and read the room.

Opening with impact

- **Lead with the customer problem, not the lab title.** The panel should feel the urgency of the problem before they hear your solution. "Imagine your customer asks..." is stronger than "Today I'll present..."
- **Make eye contact, not slide contact.** You know this material. Refer to the document occasionally, but speak to the panel as if you're having a conversation, not reading a report.
- **Set expectations early.** "I'll spend about eight minutes on the overview, then walk through each section in detail. I'm happy to go deeper on anything along the way." This gives the panel permission to interrupt, which makes the session more natural.

Collaborative whiteboarding

If the panel has a whiteboard available, consider drawing the four-box architecture diagram live instead of just referencing the document. This demonstrates teaching skill and creates engagement.

When to whiteboard
If the panel asks a question about how the layers connect, or if you sense they'd benefit from a visual. Don't force it; let it emerge naturally from the conversation.

When not to whiteboard
If you're running behind on time or the panel seems content with the document's diagrams. A whiteboard that slows things down hurts more than it helps.

Demo confidence dos and don'ts

Do

- Speak about the lab as if you've run it before
- Reference specific code samples by name
- Use phrases like "in my experience" or "what I've found works"
- Anticipate follow-up questions and address them preemptively
- Pause briefly after making a key point to let it land

Don't

- Apologize for what's not included
- Say "I would have liked to add..." without explaining why you didn't
- Rush through the reasoning section; it's where your judgment shows
- Read code blocks verbatim; summarize what they do instead
- Fill silence with filler words; embrace brief pauses

Handling deep technical questions

The "production delta" framework

When asked about production readiness or real-world deployment, use this three-step framework:

Step 1: Acknowledge the gap
"You're right that the lab uses [lab simplification]. That's a deliberate trade-off for a 60-90 minute session."

Step 2: Name the production requirement
"In production, you'd need [specific production requirement]: [brief explanation of what it adds]."

Step 3: Bridge to the lab's teaching value
"The lab teaches the core pattern and explicitly calls out the production delta so SEs can have that conversation with customers confidently."

Reading the room

Signal	What it means	How to respond
Panel member leans forward, nods	Engaged, wants more detail	Go deeper on the current point; offer a specific example
Panel member looks at the clock	Pacing concern	Summarize current section and move on: "Let me jump to the key takeaway here..."
Multiple panel members exchange glances	They have a question or concern	Pause and invite: "I see some interest—what's on your mind?"
Panel member interrupts with a question	High engagement (this is good)	Answer concisely, then bridge back: "Great question. The short answer is... Let me show you how that plays out in Section X."
Long silence after a question	They're processing or unsatisfied	Add one more layer of detail or ask: "Would it be helpful if I went deeper on that?"

Managing energy and presence

- **First three minutes set the tone.** If you start confident and clear, the panel gives you the benefit of the doubt for the rest
- **Vary your pace.** Slow down for key points, speed up for housekeeping details. Monotone kills engagement
- **Use the document as a prop, not a crutch.** "If you look at page 6, you'll see the code sample..." then look up and explain it
- **When you're nervous, focus on one panelist.** Make eye contact, deliver a point, then shift to the next person. This feels like a conversation, not a performance
- **Drink water.** Seriously. Dry mouth makes you talk faster and less clearly. Take a sip during transitions
- **Your closing is pre-written.** No matter how the Q&A goes, you end strong. That's the last impression they take away

BACKUP MATERIAL

Deep-dive extensions

Ready-to-deploy material if Q&A goes deep on specific topics. You likely won't need all of these.

Streamable HTTP vs. stdio

If asked to go deeper on transport choices

stdio: Client spawns server as a subprocess. Zero network setup, no auth needed. Great for local tools and development. Limitation: single client, server lifecycle tied to client.

Streamable HTTP: Server runs independently at a URL. Supports multiple clients, sessions, and server-initiated events. Requires auth (typically OAuth 2.1) and Origin validation. This is the production transport for remote MCP servers.

The SSE question: Server-Sent Events as a standalone transport is deprecated in favor of Streamable HTTP. If anyone mentions SSE, note that Streamable HTTP subsumes it with better session management and bidirectional communication.

Multi-tool discovery and filtering

If asked about scaling beyond one tool

In production, MCP servers expose multiple tools. The agent discovers all available tools via the `tools/list` method. Tool filtering is controlled server-side: the server can restrict which tools a given client sees based on auth context, user role, or explicit configuration. This is how enterprise governance works at the tool level.

The lab uses one tool because it teaches the full discovery-call-validate cycle. Adding a second tool (e.g., a ticketing system) doesn't introduce new concepts; it demonstrates the pattern scales. That's why it's a v3 stretch goal, not a v1 requirement.

Evaluation frameworks for MCP integrations

If asked about measuring integration quality

For POV-stage conversations, SEs can use three evaluation dimensions:

Relevance: Do the top-k results match the query intent? Measured by precision@k against a golden set of queries and expected results.

Grounding: Does the agent's answer reference returned documents? Measured by citation presence and factual alignment with source snippets.

Governance: Does the system respect permission boundaries? Measured by confirming that results with restricted `acl_group` values are filtered for unauthorized users.

BACKUP MATERIAL (CONTINUED)

More deep-dive extensions

Competitive positioning

If asked about competition

vs. Custom SDKs: Every custom SDK is a maintenance burden. MCP replaces N integrations with one protocol. The ecosystem of pre-built MCP servers reduces time-to-value.

vs. GraphQL gateways: GraphQL is a query language, not an agent protocol. MCP adds tool discovery, schema negotiation, and transport abstraction that GraphQL doesn't provide. They can coexist (MCP server backed by GraphQL).

vs. Vendor-locked connectors: Proprietary connector frameworks create lock-in. MCP is an open standard, which reduces customer switching cost and increases trust. The competitive moat isn't the protocol; it's the implementation quality and ecosystem breadth.

Self-service lab kit design

If asked about scaling without facilitators

The self-service kit would include:

GitHub repo: All lab code, including the mock API Lambda, MCP server, and validation scripts. README with step-by-step instructions and screenshots.

Infrastructure-as-code: Terraform or CDK template that provisions the Lambda, IAM role, and CloudWatch log group in one command. Teardown script included.

Self-check script: A validation script that confirms each lab section was completed correctly (endpoint responds, tool is discoverable, query returns grounded answer). SEs run it at the end to confirm success.

Video companion: 15-minute recorded walk-through of the facilitator completing the lab. SEs can reference it when stuck without waiting for a live session.

Authorization model deep-dive

If security team questions go deep

MCP's authorization model for HTTP transports follows OAuth 2.1 patterns. The server acts as a resource server, the client presents tokens scoped to specific tools and data sets. Key points:

Origin validation: HTTP transport servers validate the Origin header to prevent unauthorized cross-origin tool calls.

Tool-level scoping: Servers can expose different tool sets to different authenticated clients, enabling least-privilege access.

Audit trail: Every tool call passes through the MCP layer with structured request/response logging, providing a natural audit surface.

EMERGENCY REFERENCE

Panic sheet

"If X happens, do Y." Quick reference for common delivery mishaps.

If you lose your train of thought mid-sentence
Pause. Take a breath. Say: "Let me reframe that." Then start the point fresh. A brief pause reads as thoughtfulness, not confusion.

If a panel member asks something you genuinely don't know
Say: "That's a great question, and I want to be honest that I don't have a complete answer. Here's how I'd go about finding one: [describe process]. What I can tell you is [bridge to something related you do know]."

If you're running out of time and still have material left
Skip to your closing statement. A strong close matters more than complete coverage. Say: "In the interest of time, let me jump to the key takeaway..." and deliver your prepared close.

If a panel member challenges a design decision aggressively
Don't get defensive. Acknowledge: "That's a valid concern. The reasoning was [explain]. If I were optimizing for [their concern], I'd adjust by [alternative]. What I prioritized here was [your priority] because [reason]."

If the panel seems disengaged or distracted
Ask a direct question: "Before I move on, is there a specific area you'd like me to go deeper on?" This resets engagement and gives you information about what they care about.

If technology fails (screen share drops, document won't open)
Stay calm. You know the material. Say: "Let me describe what you'd be seeing on page X..." and continue verbally while troubleshooting. The panel is evaluating you, not your screen share.

If the panel asks the same question twice (or a variation)
They didn't get a satisfactory answer the first time. Reframe your answer with a different angle or a concrete example. "Let me come at that from a different angle..."

If you realize you made an error in something you said
Correct it immediately: "Actually, let me correct myself on that point..." Self-correction shows intellectual honesty and attention to accuracy. It's a positive signal.

CHECKLIST

Pre-session checklist

Complete before the panel session begins.

Day before

- ✓ Review the full lab design document (14 pages) once through
- ✓ Read through this presenter notes document once
- ✓ Practice the opening hook out loud (30 seconds)
- ✓ Practice the closing statement out loud (30 seconds)
- ✓ Identify your 3 strongest talking points to emphasize
- ✓ Review the Anticipated Q&A section and rehearse 5 answers
- ✓ Test screen sharing with the PDF document
- ✓ Confirm video, audio, and internet connection quality

30 minutes before

- ✓ Close unnecessary applications and browser tabs
- ✓ Open the lab design PDF and presenter notes PDF side by side
- ✓ Have water within reach
- ✓ Silence phone and notifications
- ✓ Take five slow breaths to settle your nervous system
- ✓ Remind yourself: you know this material deeply

During the session

- ✓ Start with the customer problem, not the lab title
- ✓ Watch the clock at 8:00, 28:00, and 40:00 (transition points)
- ✓ Keep Q&A answers to 60-90 seconds
- ✓ Bridge back to the lab's value after each question
- ✓ Deliver the prepared closing statement no matter what

Timing trigger points

Time	You should be at	If behind
8:00	Finishing Part 1 (overview), transitioning to walk-through	Drop the bridge statement, go straight to section details
16:00	Finishing Sections 3-4 (API + MCP wiring)	Summarize Sections 5-6 instead of walking through detail
28:00	Finishing walk-through, opening Q&A	If you haven't covered reasoning, work key trade-offs into Q&A answers
40:00	Wrapping Q&A, moving to close	Deliver closing statement immediately regardless of question queue

DELIVERY SUMMARY

What a great delivery looks like

If the panel walks away thinking these things, you've succeeded.

- ✓ "They understand MCP at a level that would impress our engineering team"
- ✓ "They designed for the SE learner, not for themselves"
- ✓ "Every technical decision connected back to a customer outcome"
- ✓ "They made smart trade-offs and could explain why"
- ✓ "They handled tough questions with poise and honesty"
- ✓ "They think about enablement as a program, not a one-off session"
- ✓ "I'd trust them to run our technical enablement"

> "One protocol. Many enterprise systems. Grounded answers. This lab gives SEs the hands-on confidence to demo, position, and troubleshoot MCP integrations with real customer impact."

Prepared by John Yzaguirre
Technical Enablement Manager Assignment