

COMPANION WORKSHEET

Chapter 6 — Platform Lane Decision

Use a fresh copy from the Companion Package. This worksheet chooses one primary platform lane and one fallback. It does not recommend a product, authorize a purchase, prepare a host, install software, prove acceleration, or predict performance.

A. Record the inherited evidence

Date and local time

Non-sensitive system label

Chapter 4 hardware-inventory path

Chapter 5 model-memory-fit path

Platform evidence folder

Carry forward the Chapter 5 rows without silently changing artifact, quantization, context, loaded-model count, parallel-request count, topology, or verdict.

Candidate role	Exact artifact/variant	Quantization	Target context	Concurrent models	Parallel requests	Chapter 5 topology	Chapter 5 verdict	Evidence path
Conservative								
Working								
Stretch								

B. Define relevant platform lanes

A lane is an exact path, not a brand or operating-system name. Evaluate only lanes relevant to owned or contemplated hardware.

Lane label	Host OS/version family	Accelerator or CPU class	Driver/compute interface	Runtime/deployment path	Artifact identity / compatibility	Intended placement

C. Separate documentation status from edition testing

Allowed documentation status values only:

Supported

Conditional/experimental

Not documented

Contradicted

Allowed edition-test status values only:

Tested by this edition

Not tested by this edition

Not applicable

Official documentation and edition testing answer different questions. Never convert community success into official support or official support into book-tested proof.

Conditional/experimental never becomes Supported. It may continue provisionally only when its named condition is explicitly accepted as a decision constraint and an exact Chapter 9 proof target is bound. Otherwise mark the official-support planning filter Unresolved and defer the lane.

Lane label	Documenta tion status	Official authority and URL	Access date	Exact versi on/device/h ost qualific ation	Edition-test status	Edition evidence path or reason

D. Apply hard filters

Use Pass, Reject, Unresolved, or Not applicable for planning filters only. Record the later proof as Pending, Observed – matches, or Observed – contradicts. Pending is not an unresolved planning gate: it permits a provisional choice only after every planning filter passes, and it never proves placement or acceleration.

Lane label	Owned-h ardware match	Chapter 5 candid ate fit	Artifact compati bility	Official-s upport b oundary	Expecte d place ment basis	Operatio nal path understo od	Chapter 9 proof status and exact target	Result and reason

E. Record placement and multiple-device assumptions

Do not add device-memory totals and call them one pool. Describe the expected placement and its evidence.

Lane label	Physical memory pools	Per-device capacity	Aggregate installed capacity, if relevant	Expected model placement or split	Documenta tion/eviden ce	Later observation required

Multi-device conclusion:

What this evidence does not prove:

F. Choose the primary lane

Primary lane label

Chapter 5 candidate used

Exact lane tuple

Why this lane survives every hard filter

Official documentation status and qualification

Edition-test status

Expected placement

Chapter 9 proof status (Pending expected here)

One exact Chapter 9 observation target

Chapter 9 acceptance criteria

Invalidation condition

G. Design a fallback independent of the named primary failure

The fallback must remain credible when the named primary dependency fails. If its workload changes, reference a recalculated Chapter 5 row; do not inherit the old fit verdict.

Primary failure covered

Fallback lane label

Dependency removed or changed

Chapter 5 candidate used

Reduced workload assumptions, if any

Recalculated Chapter 5 verdict/evidence path, if required

Why the fallback does not share the failed dependency

Official documentation status and qualification

Edition-test status

Expected placement

Chapter 9 proof status (Pending expected here)

One exact Chapter 9 observation target

Chapter 9 acceptance criteria

Invalidation condition

Fallback independence test

Compare the dependency that failed in the primary lane with the fallback route. Use only Independent, Shared but not failed dependency, Not independent, or Unresolved in the last column. A fallback cannot pass while the named failed dependency is Shared but not failed dependency, Not independent, or Unresolved.

Dependency	Primary lane value/evidence	Fallback lane value/evidence	Independence result
Host OS/version family			
Driver/compute interface			
Runtime/deployment path			
Artifact/format compatibility			
Memory topology and required pool			
Candidate, context, loaded-model count, and parallel requests			

Named failed dependency

Independence conclusion and evidence

H. Record rejected or deferred lanes

Lane label	Rejected or Deferred	First failed or unresolved gate	Evidence	What could reopen it

I. Stop/Go record

- ☐ Exactly three Chapter 5 candidates were carried forward without silent changes.
- ☐ Every evaluated lane is an exact host/device/interface/runtime/artifact/placement tuple.
- ☐ Documentation status and edition-test status use separate finite fields.
- ☐ Every support claim has an official authority, access date, and qualification.
- ☐ Multiple-device capacity is not presented as one transparent memory pool without exact evidence.
- ☐ Exactly one primary lane survives every recorded hard filter.
- ☐ A Pending Chapter 9 proof is separate from every planning filter and is not presented as observed proof.
- ☐ Exactly one fallback remains credible when the named primary dependency fails.
- ☐ Any changed fallback workload points to a recalculated Chapter 5 verdict.
- ☐ Primary and fallback each name one exact Chapter 9 observation target, a Pending proof status, and an invalidation condition.
- ☐ Neither choice is presented as purchase approval, completed installation, proven acceleration, or measured performance.

Primary platform decision:

Fallback decision:

What Chapter 7 may evaluate:

What Chapter 9 begins to prove: host/device/interface visibility and lane-appropriate memory

What later chapters must finish: Chapter 10 container-device visibility if selected; Chapter 11 exact runtime, artifact, and observed placement

What could reopen this decision: