Client → Proxy

Req → Route → DAP Handler

↓

JSON RPC
Handler

---

JSON RPC
Request Handler → Read & Modify
Client → Delve
↳ Parse Request
↳ Handle based on req Method

RPC Server. Eval
RPC Server. List Local Vars
RPC Server. List Function Args
RPC Server. Command → Extract Cmd Name from req. Params
↳ Track Cmd Name for Response Correlation

} Translate Frame
↳ Get Frame Mapping from Response Handler

---

Response Handler → Read & Modify
Delve → Client
↳ Parse Response from Delve → RPC Server. Stack Trace → Filter Stack Trace
Response
} Response to StackTrace Request

Body contains
Locations
File
Line

↳ RPC Server. Command. * → RPC Server. Command. Next
RPC Server. Command. continue

Save some of Debugger & state
Current Thread's File Location
File Line
} State
Current Thread's Function Name
User stepped in external breakpoint — Yes → Repeated Step-Over until reach workflow code
↳ Loop:
Max iterations = 30
Step Over
Update State
↳ Reached Workflow code? — YES → Track & new UX Step ID
the original Cmd from Client
is StepOver Command
↳ NO → Break

---

Important Parts:
1 - How to hide non-workflow code in stackTrace response?
2 - How to know the debugger stopped at workflow code?

} IDE get the correct information from the debugger

3. How to do "internal" step-over?
} Always stay in workflow code

---

Solutions
1 Maintain a frame mapping between req & resp handler
2 Use current file path in debugger state
3 Use a special ID range for internal stepover request the debugger sends to Delve.