MCP의 SDK 간 프로토콜 적합성 분석

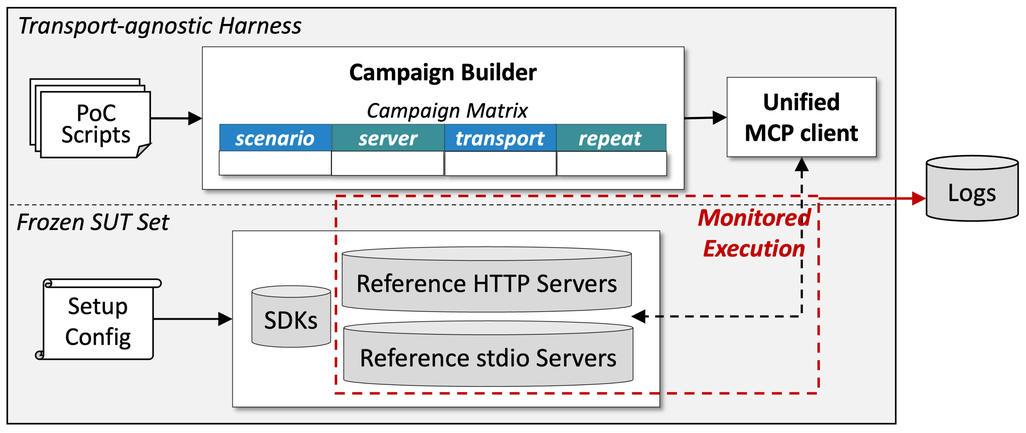
Cross-SDK Protocol Conformance Analysis of MCP

Ein Kim, Seungsoo Lee
Incheon National University

초 록

- MCP는 LLM과 외부 자원을 연결하는 표준 프로토콜이지만, 공식 SDK 구현의 명세 준수 여부에 대한 체계적 검증이 부족함.
- 본 연구는 7개 공식 SDK의 stdio 및 Streamable-HTTP 전송을 대상으로 적합성 검증을 수행함.
- **서버 비정상 종료**, **세션 상태 변조**, **응답 의무 명세 미준수의** 3개 결함 범주를 분석함.
- 7개 SDK 중 6개에서 하나 이상의 결함을 발견했으며, 결과를 책임 있는 공개 절차에 따라 SDK 관리자에게 보고함.

- 테스트베드 디자인



실험 설계

SDK	Package	Version	stdio Server	HTTP Server
TypeScript	`@modelcontextprotocol/sdk`	$\leq$ 1.27.1	`server-everything`	`server-everything`
Python	`mcp` (pip)	$\leq$ 1.26.0	`mcp-server-fetch`	`everything-server`
Go	`go-sdk` (Go)	$\leq$ 1.4.1	`everything`	`everything`
Rust	`rmcp` (Rust)	$\leq$ 1.2.0	`counter_stdio`	`counter_streamhttp`
Ruby	`mcp` (RubyGems)	$\leq$ 0.9.0	`stdio_server`	`http_server`
Kotlin	`kotlin-sdk` (Maven)	$\leq$ 0.9.0	`kotlin-mcp-server`	`simple-streamable`
PHP	`mcp/sdk` (Packagist)	$\leq$ 0.4.0	`discovery-calculator`	`discovery-calculator`

- TypeScript, Python, Go, Rust, Ruby, Kotlin, PHP의 7개 공식 SDK와 예제 서버를 사용함.
- stdio와 HTTP 전송 방식별 응답, 오류 코드, 세션 상태 변화를 기록함.

아래 범주에 대한 테스트 시나리오를 각 10회 반복하여 재현성을 확인함.

- **(심층 중첩 입력)** 깊게 중첩된 JSON 메시지를 입력하여 서버 생존 여부를 확인함.
- **(중복 초기화)** 초기화 완료 후 운영 단계에서 초기화 요청을 재전송하여 세션 상태 변조 가능성을 확인함.
- **(응답 의무)** id가 포함된 요청에 대해 결과 또는 오류 응답을 적절히 반환하는지 확인함.

심층 중첩 입력에 의한 자원 고갈

- **HTTP 구현은 프로세스 종료가 관찰되지 않음.**

SDK	Depth	Size	Outcome
Rust	128	1,218 B	Process exit (code 0)
Python	256	2,370 B	Process exit (code 1)
Ruby	256	2,370 B	Timeout (recovers)
Go	10,000	99,066 B	Process exit (code 0)
PHP	–	–	Error −32700
Kotlin	–	–	Error −32603
TypeScript	–	–	OK at 100k depth

- Rust, Python, Go는 stdio에서 프로세스 종료가 발생함.
- Ruby는 stdio에서 타임아웃 후 스스로 회복함.
- TypeScript는 100k depth 입력도 정상 처리함.
- PHP와 Kotlin은 오류 응답을 반환하며 프로세스 가용성을 유지함.

중복 초기화에 의한 세션 상태 변조

- **7개 중 6개 SDK가 하나 이상의 전송 방식에서 중복 초기화를 수락함.**

SDK	stdio	HTTP	Transport-Indep.
TypeScript	✗ Accepted	✓ Rejected	No
Python	✗ Accepted	✓ Rejected	No
Go	✗ Accepted	✗ Accepted	**Yes**
Rust	✗ Accepted	✗ Accepted	**Yes**
Ruby	✗ Accepted	✓ Rejected	No
Kotlin	✗ Accepted	✓ Rejected	No
PHP	✓ Rejected	✓ Rejected	N/A

- 초기화 협상 완료 이후에도 프로토콜 버전과 기능 집합이 변경될 수 있음.
- Go와 Rust는 두 전송 모두 수락함.
- PHP만 두 전송 모두 거부함.
- TypeScript, Python, Ruby, Kotlin은 stdio에서는 수락/ HTTP에서는 거부함.

응답 의무 명세 미준수

- **HTTP 구현은 모두 정상 응답함.**

Tier	Test Message	TS	Py	Go	Rust	Ruby	Kotlin	PHP
T1	Wrong `jsonrpc` version	✗	✗	✗	✗	✗	–	✗
T1	Missing `jsonrpc` field	✗	✗	✗	✗	✗	–	✗
T1	Method as number	✗	✗	✗	✗	✗	–	✗
T1	Null params	✗	✗	✗	✗	–	–	–
T1	Missing required params	–	–	–	–	–	–	–
T1	Extra top-level field	✗	–	–	–	–	–	–
T1	`__proto__` injection	–	✗	✗	✗	–	–	–
T2	Params as array	✗	✗	✗	✗	–	–	–
T2	Id as null	–	–	–	–	–	–	–
T3	Empty method string	–	✗	✗	✗	–	–	✗
T3	Non-existent method	–	✗	✗	✗	–	–	✗
T3	Id as string	–	✗	✗	✗	–	–	–

✗ = silent drop; – = proper error response; T1 = structurally malformed. T2 = MCP-specific restriction; T3 = semantically valid input.

- stdio 구현 6개에서 응답 의무 미준수가 관찰됨.
- 일부 SDK는 JSON-RPC 2.0상 유효한 문자열 id 요청도 폐기함.
- 유효 요청의 무응답은 클라이언트 무한 대기 및 서비스 거부 조건을 형성할 수 있음.

결 론

- MCP 공식 SDK 구현에서 서버 가용성, 세션 무결성, 응답 의무 측면의 적합성 결함을 발견했으며, 명세에 중복 초기화 거부 의무 조항 추가를 제안함.
- **모든 결함 범주에서 stdio가 Streamable-HTTP보다 낮은 적합성을** 보이며 전송 계층의 오류 격리 방식이 MCP 서버 안정성에 직접적인 영향을 미침.
- 향후 적합성 및 취약성 테스트를 자동화하는 MCP 프로토콜 침투 테스트 프레임워크로 확장할 예정임.