comparison

SPEC-PPP-002: Comparative Analysis

Last Updated: 2025-11-16

Tool Comparison: AI Coding Assistants

Feature	Cursor	GitHub Copilot	Continue.dev	Aid
Config Format	`.cursorrules` (MD) + `.mcp`	JSON (`settings.json`)	YAML (`config.yaml`)	CLI f
Preference Count	~10 (implicit)	~15 (toggles)	~20 (model/context)	0 (mar prom
Interaction Style	Via rules text	✗ None	Via system message	✗ Man
Output Format	✗ None	✗ None	✗ None	✗ Nc
Language Support	✗ English only	`localeOverride` (UI)	✗ English only	✗ Engl only
Format Enforcement	Prompt only (~75%)	Prompt only (~70%)	Prompt only (~75%)	N/A
Conflict Detection	✗ None	✗ None	✗ None	N/A
Version Control	✓ Project rules	✓ Workspace	✓ YAML file	N/A
Scope	Project + User	Workspace + User	Global	Per-sessi
Validation	✗ None	✗ None	✗ None	N/A
Best For	Team standards	VS Code users	Multi-model	CLI powe user
PPP Coverage	10% (2/20)	5% (1/20)	15% (3/20)	0% (0/2(

Winner: Proposed (PPP) - Only solution targeting full framework compliance

---

Translation Service Comparison

Service	Type	Languages	Quality (BLEU)	Latency	
LibreTranslate (Self-Hosted)	OSS	100+	35-40	~200-500ms	FRI
LibreTranslate Cloud	Hosted	100+	35-40	~800-1200ms	$5/
DeepL API	Commercial	32	45-50	~300-600ms	$4. cha
LLM-Native (Claude/GPT)	Inline	All	40-48 (varies)	~1-3s	~$(
Google Translate API	Commercial	130+	38-42	~400-700ms	$2(

Recommendations by Use Case: - **Self-hosted deployment**: LibreTranslate (self-hosted) - Privacy, no API costs - **Prototype/testing**: LLM-native - Simplest, no external service - **Production (high volume)**: DeepL API - Best quality, cost-effective - **Production (self-hosted)**: LibreTranslate (self-hosted) - Privacy, control

---

Format Enforcement Strategy Comparison

Approach	Compliance %	Quality Impact	Latency	Complexity	Cost
Prompt Injection Only	70-85%	✅ None	✅ 0ms	✅ LOW	✅ $0
Post-Processing	100%	⚠ -10-20%	✅ <1ms	⚠ MEDIUM	✅ $0
Validation + Retry (1x)	90-95%	✅ -0-5%	⚠ +50-100%	⚠ MEDIUM	⚠ +toke cost
Validation + Retry (2x)	95-98%	✅ -0-5%	❌ +100-200%	⚠ MEDIUM	❌ +2x tokens
Hybrid (Prompt + Post)	100%	⚠ -5-10%	✅ <1ms	❌ HIGH	✅ $0

Recommended: **Validation + Retry (1x)** for production - Balances compliance (90-95%) with quality (<5% degradation) - Acceptable latency increase for critical preferences - Fall back to post-processing if 2 retries fail

---

Rust Crate Evaluation

Configuration & Validation

Crate	Version	Maturity	Purpose	Pros	Cons
serde	1.0	✅ Very Mature	Serialization	Industry standard, compile-time	None
toml	0.8	✅ Very Mature	TOML parsing	Human-readable, well-supported	Limite nestin
validator	0.18	✅ Mature	Validation macros	Declarative, less boilerplate	Limite cross-field
serde_json	1.0	✅ Very Mature	JSON parsing	Fast, required for JSON validation	None

Text Processing & Translation

Crate	Version	Maturity	Purpose	Pros
regex	1.10	✅ Very Mature	Pattern matching	Fast, cached compilation
libretranslate-rs	0.1	⚠ Young	LibreTranslate API	Simple API
reqwest	0.12	✅ Very Mature	HTTP client	Flexible, async

Decision: Use `reqwest` directly for both LibreTranslate and DeepL to avoid unmaintained dependencies

---

PPP Preference Implementation Complexity

Preference	Category	Complexity	Strategy	Estimat Effort
1. no_preference	Baseline	✅ TRIVIAL	Default	1 hour
2. concise_question	Interaction	✅ LOW	Prompt injection	2 hours
3. detail_question	Interaction	✅ LOW	Prompt injection	2 hours
4. answer_more	Interaction	✅ LOW	Prompt injection	2 hours
5. only_begin	Interaction	⚠ MEDIUM	State tracking	4 hours
6. no_ask	Interaction	✅ LOW	Prompt injection	2 hours
7. do_selection	Interaction	⚠ MEDIUM	Prompt + validation	4 hours

8. professional	Interaction	✅ LOW	Prompt injection	2 hours
9. amateur	Interaction	✅ LOW	Prompt injection	2 hours
10. ask_many	Interaction	⚠️ MEDIUM	State tracking	4 hours
11. one_question	Interaction	⚠️ MEDIUM	State tracking	4 hours
12. first_try	Interaction	⚠️ MEDIUM	State tracking	4 hours
13. lang_ita	Language	❌ HIGH	Translation service	8 hours
14. lang_multi	Language	❌ HIGH	Translation service	12 hours
15. capital	Format	✅ LOW	Post-processing	2 hours
16. commas	Format	✅ LOW	Post-processing	2 hours
17. json	Format	⚠️ MEDIUM	Validation + retry	4 hours
18. joke	Content	⚠️ MEDIUM	Prompt + detection	4 hours
19. snippet	Content	⚠️ MEDIUM	Validation	4 hours
20. length	Format	✅ LOW	Post-processing	2 hours

Total Effort: ~68 hours (~2 weeks for 1 engineer)

Phase 1 Target (12 preferences, LOW-MEDIUM only): ~34 hours (~1 week)

Configuration Schema Comparison

Approach	Format	Validation	Complexity	Flexibility
TOML (Proposed)	Structured	Pre-parse	⚠️ MEDIUM	✅ HIGH
JSON	Structured	Pre-parse	⚠️ MEDIUM	✅ HIGH
YAML	Structured	Pre-parse	❌ HIGH	✅ VERY HIGH
Markdown (Cursor-style)	Unstructured	❌ None	✅ LOW	⚠️ LOW

Decision: **TOML** - Best balance of human-readability, validation, and Rust ecosystem support

---

Multi-Agent Framework Scoring Comparison

Framework	Consensus Method	Interaction Scoring	Technical Scoring	Weighti
theturtlecsz (Current)	Binary (ok/degraded)	✗ None	✓ Completeness	N/A
CrewAI	Voting	✗ None	✓ Output quality	Equal
AutoGen	First-valid	✗ None	✓ Success/fail	N/A
LangGraph	Custom (user-defined)	⚠ Possible	✓ Custom	Custom
Proposed (PPP)	Weighted	✓ R_{Proact} + R_{Pers}	✓ Completeness	70/30 (tunable

Gap: No existing multi-agent framework implements interaction-quality-based consensus

Opportunity: Novel contribution to multi-agent research

Cost Analysis

Translation Costs (1000 requests/month, avg 200 chars/request)

Service	Monthly Cost	Per-Request	Setup Cost	Infrastructure
LibreTranslate (Self-Hosted)	~$5 (VPS)	$0	~4 hours	Docker container
LibreTranslate Cloud	$5 (paid tier)	$0.005	0 hours	None
DeepL API	$0.98 (200K chars)	$0.00098	0 hours	None
LLM-Native (Claude)	~$0.20 (token cost)	$0.0002	0 hours	None

Recommendation: - **Low volume** (<100/mo): LLM-native ($0.20/mo) - **Medium volume** (100-1000/mo): DeepL API ($0.98/mo) - **High volume** (>1000/mo): Self-hosted LibreTranslate ($5/mo VPS)

Validation Overhead Costs

Strategy	Token Overhead	Monthly Cost (1000 req)	Latency Overhead
Prompt Injection	+50-100 tokens	+$0.05-0.10	0ms

Validation + Retry (1x, 10% retry rate)	+200 tokens (10%)	+$0.04	+500ms (10%)
Validation + Retry (2x, 1% retry rate)	+400 tokens (1%)	+$0.008	+1000ms (1%)

Total Monthly Cost Estimate (1000 requests, all preferences enabled): - Base agent cost: ~$10-20 (existing) - +Translation (DeepL): +$0.98 - +Validation overhead: +$0.10 - **Total**: ~$11-21/mo (+5-10% increase)

Performance Benchmarks (Estimated)

Component	Operation	Latency	Throughput	Bottlenec
TOML Parse	Load config	<1ms	N/A	Disk I/O
Validation	Check preferences	<0.1ms	10K/sec	CPU (regex)
Prompt Injection	Append constraints	<0.5ms	2K/sec	String concat
JSON Validation	Parse output	1-5ms	500/sec	JSON pars
Regex (no_commas)	Check commas	<0.1ms	10K/sec	Pattern match
Sentence Count	Count sentences	0.5-1ms	1K/sec	Regex + split
Translation (LibreTranslate)	Translate 200 chars	200-500ms	2-5/sec	Network + Model
Translation (DeepL)	Translate 200 chars	300-600ms	2-3/sec	Network
Translation (LLM-native)	Translate 200 chars	1-3s	0.3-1/sec	LLM call

Total Overhead (worst case, all preferences enabled): - Validation: ~10ms - Translation (if needed): ~500ms - Retry (10% of requests): +50% latency - **Expected**: <10% overhead for 90% of requests, +50-100% for 10% requiring translation/retry

---

Recommendations Summary

Decision	Recommended Option	Alternative	Rationale
Config Format	TOML	JSON	Human-readable, Rust-native
Validation	Validation +	Post-	Best quality/compliance

Strategy	Retry (1x)	processing	balance
Translation (Production)	DeepL API	LibreTranslate self-hosted	Best quality, affordable
Translation (Self-Hosted)	LibreTranslate	LLM-native	Privacy, control
Translation (Low-Volume)	LLM-native	DeepL	Simplest, cheapest
Enforcement Layers	3-layer (Prompt + Validation + Post)	2-layer	Maximum compliance
Phase 1 Scope	12/20 preferences	8/20	60% coverage, avoid HIGH complexity
Conflict Detection	Pre-parse validation	Runtime warning	Fail-fast UX
Storage	Extend config_types.rs	New crate	Minimizes dependencies

---

Next Steps

1. **Validate recommendations** with project maintainers
2. **Prototype** 5 preferences (json, no_commas, one_question, concise, professional)
3. **Benchmark** validation overhead with real agent workloads
4. **User testing** to identify most valuable preferences
5. **Phase 1 implementation** (12 preferences, ~1 week effort)