Magmide

Bringing a world of unhackable unbreakable software.

Broken software costs
tens of trillions of dollars
every year.

Theft, damage, churn, regulatory penalties.

Software doesn’t have to be broken!

It can be *proven correct*.

Proven correct?

Write code, including formal spec and proof…

Verify using *proof checking algorithm*.

Recent breakthroughs make *mainstream verification possible*.

New paradigms, AI can automate more proofs.

Verified software is becoming *imperative*.

Generative AI unsafe without it.

Generative AI can automate hacking.

Software ever more critical part of society.

Existing proof languages are garbage.

Slow, dogmatic, academic, obtuse, limited.

Magmide designed for working engineers.

Fast, realistic, reusable, ergonomic, clear.

We’re building Magmide language.

Targeting open source adoption in the short term.

Many possible business models.

Proofs-as-a-service. Given code and formal spec, API returns proof *(most of the time)*.

Proofs-*and-code*-as-a-service. What if copilot gave provably correct results?

No-code tools and cloud databases. Airtable/Salesforce/etc but all verified.

Hardware with verified firmware. Something like Oxide Computer.

Mainstream verification would *transform software engineering*.

TAM trillions of dollars.

No competitors.

No one is targeting mainstream verification.

Only consultancies or niche/impractical tools.

Blaine Hansen

founder/CEO

Staff Engineer at MarketDial

Tej Chajed

founding technical advisor

MIT PhD

Assistant Professor UW-Madison

Need $500,000 (24 months) to build self-sustaining open source language.

Will experiment and find market fit.

Will try new open source monetization concepts to extend runway.

Broken software costs

- cybersecurity losses projected at $10.5T globally every year by 2025 [1]
- estimated $1.5T due to operational failures in 2020, just the US [2], estimated rise to $1.8T in 2022 [3]
- estimated 3.6B people affected and $1.7T lost by software failures 2017 [4]
- UAV executed kill without human confirmation [5]
- DNS outage cost Facebook ~$50M [6]

Regulatory/social/tech trends

- NSA urges shift to safe programming languages [7]
- memory safety research included in appropriations bill [8]
- Consumer Reports publishes exposé on unsafe languages [9]
- $2.7B in GDPR fines issued to date [10]
- FTC sues Chegg over data breaches [11]
- Australia increases privacy breach maximum penalty to AUD 50M [12]
- "end of Moore's law" causing more cores/software in more places [13]

Formal verification

- verified software slowly deployed in military and infrastructural applications [14] [15]
- Iris Separation Logic research brings new scalable paradigms [16] [17]
- AI can be used to automate much more theorem proving [18] [19] [20]
- "trackable invariants" and/or capability systems can enforce constraints and prevent attacks, especially well-understood vectors like sql injection, XSS, buffer overflow, secret leaking [21] [22]

Market size

TAM at least existing security/reliability spend.

- cybersecurity $172B in 2022 [23], expected $219B in 2023 [24]
- cyber insurance $11B in 2022, estimated $29B by 2027 [25]
- software monitoring estimated $3.5B in 2023 [26]
- software testing $45B in 2022 [27]

We think TAM is a percentage of total software spend.

- McKinsey estimates cybersecurity actually $2T, current market underpenetrated [1]
- total software spend $794B in 2022 [28]
- 35% of IT budgets on QA and testing [29]

Also expect Magmide would expand entire software market by increasing security/reliability ROI.

Initial beachhead

Plan to initially target sophisticated and risk averse companies with high failure costs but no mature verification standards.

- infrastructural software (google, amazon, microsoft, cloudflare): $136B in 2021 [30]
- financial services (square, stripe, amex, goldman sachs): $118B in 2021 [31] [32]
- hardware/firmware (apple, samsung, dell): firmware $21.5B by 2027 [33]
- industrial control: $172B in 2022 [34]

Also open source monetization strategies [35]

Underpowered/narrow tools

- Kani: not full proof checker
- Liquid Haskell: not full proof checker, not bare metal
- Ivy: not a full proof checker (only first order)
- Rudra: not full proof checker
- Prusti: not full proof checker
- RustHorn: not full proof checker
- KLEE and related tools: not full proof checker, only generates tests
- Vale: niche tool only suitable for cryptography verification
- Trust-in-Soft: not full proof checker
- TLA+: not a full proof checker (not based on dependent type theory), only a specification language without programming capability
- Isabelle/HOL, ACL2, PVS, Twelf: not maxed out in logical power, missing either dependent types or higher order logic or separated proposition types
- Dafny: not bare metal
- Rodin tool/B-method: not a full proof checker (only first order), unscalable (doesn't use separation logic)
- Questa: not full proof checker
- DAML: not full proof checker, blockchain focused
- Certik Move: not full proof checker, blockchain focused
- Tezos: blockchain focused, relies on obtuse academic tool (Coq)
- Yatima: blockchain focused, forces impractical functional paradigm

Deeper comparison to other tools

Obtuse academic/unrealistic tools

- Coq: forces impractical functional paradigm, allows extremely confusing custom syntax, doesn't use bare metal programming or metaprogramming, "extraction" to runnable code sloppy and tacked on, generally performs poorly
- F* (pronounced F star): forces impractical functional paradigm, uses muddled "functional effects" concept, doesn't use bare metal metaprogramming
- Lean: forces impractical functional paradigm, allows extremely confusing custom syntax, doesn't use bare metal programming
- DeepSpec: only verifying extant systems, forces painful old paradigms, not intended/suited for mainstream adoption
- ATS: difficult design, impenetrable documentation, forces impractical functional paradigm, painful requirement to only use subset types

Non-competitor companies

- Galois: consultancy, not building mainstream tools
- Synopsys: consultancy, chip verification not software verification
- Digamma: consultancy, using AI for "normal" applications like threat detection, not building general purpose tool
- Informal Systems: consultancy, blockchain focused
- Hacken: consultancy, blockchain focused
- Veriflow: only using existing tools to verify network flows, not building mainstream tools, *acquired by VMware*
- Bedrock Systems: only using existing tools to build verified virtual machine, not building mainstream tools