Will machines dream of λ calculus?

Functional Programming for the Age of AI

KC Sivaramakrishnan
1st October 2026
CSE, IITM





Qualifications and Experience

- **Assistant Professor (IITM)**
  - 2019 to 2022, 2025 — Present
- **CTO @ Tarides (2022-2024)**
  - Adjunct Professor (IITM)
- **Senior Research Fellow, University of Cambridge (2014-2018)**
  - 1851 Royal Commission and Darwin College, Cambridge, Research Fellowships
- **PhD (2014) and MS (2011), Purdue University**

Research goal

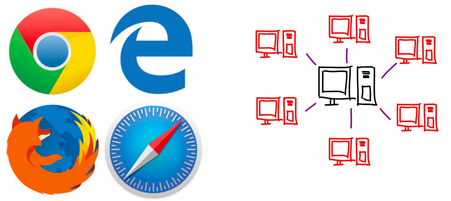
λ *Functional Programming!*

& proof-oriented PLs

Develop programming language abstractions and tools that empower developers to build secure, scalable and reliable software systems.

& agents

Secure-by-construction:
Avoid bugs that could be exploited by attackers
(agentic)

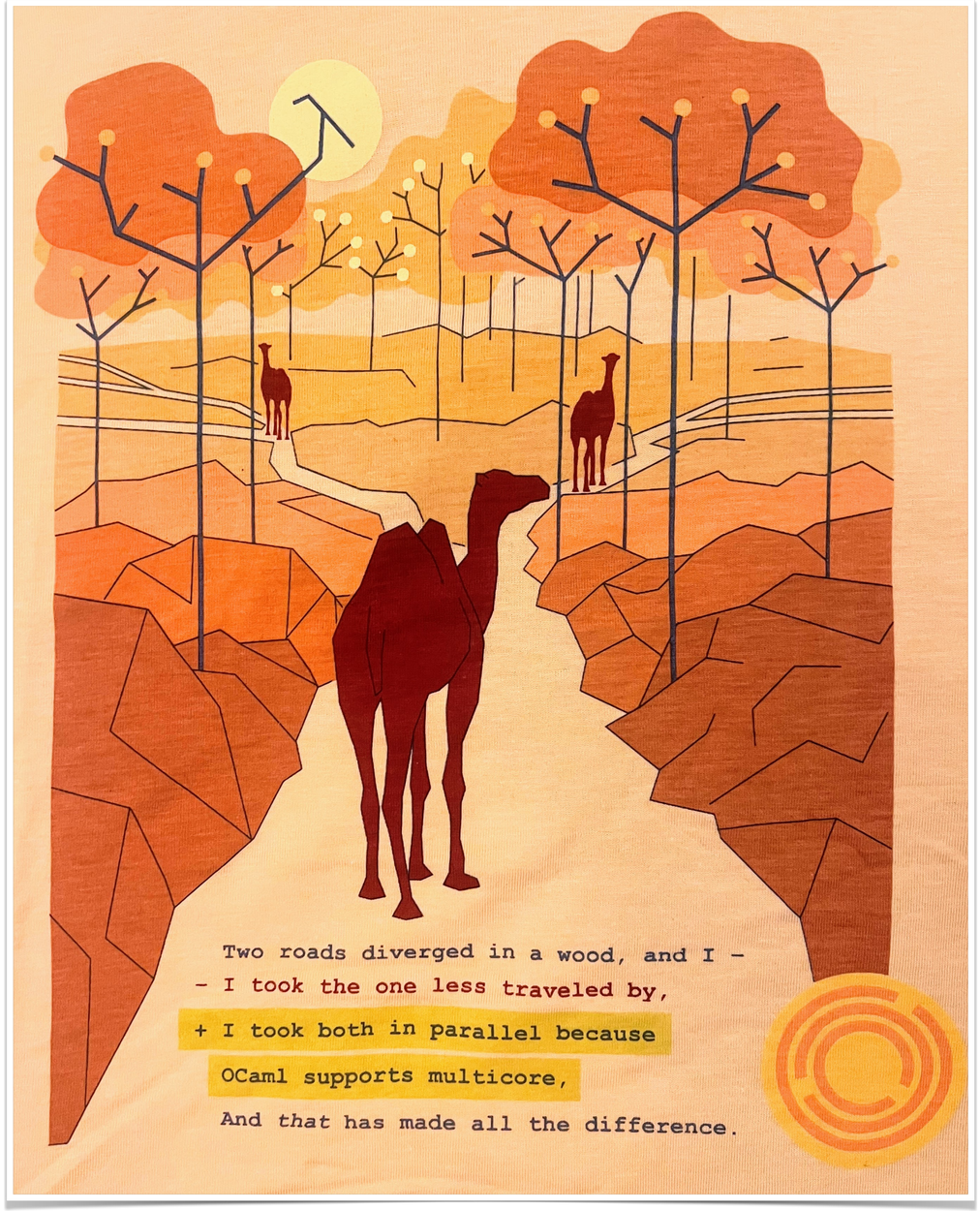




Tolerate failures in network, process, consistency

Research Highlights (1)

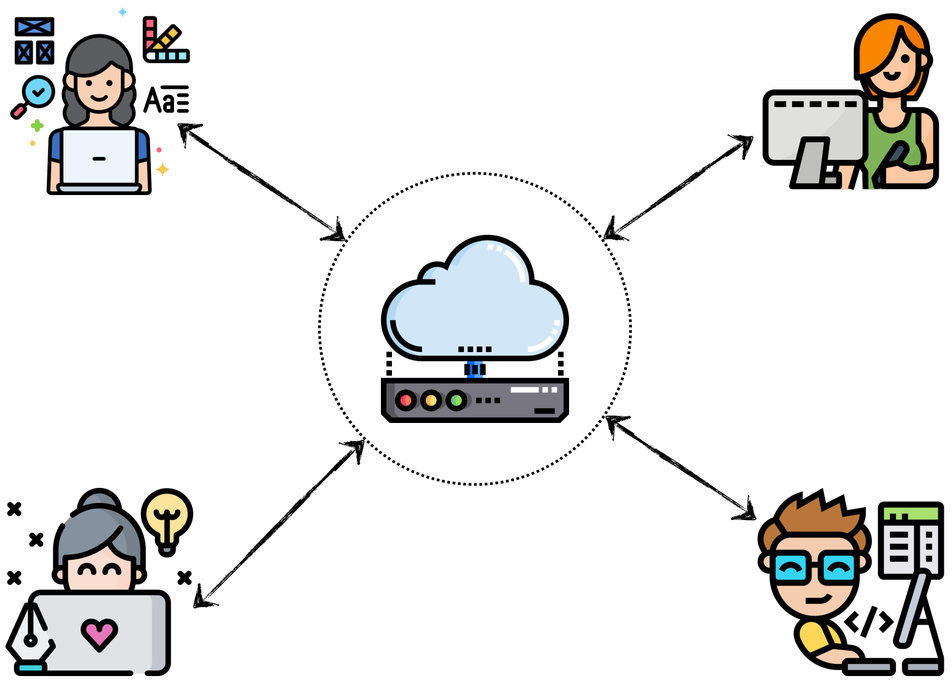
- **Multicore OCaml**
  - Concurrency and parallelism in OCaml
    - Language and runtime design and implementation
  - Papers — ***ICFP 2020*** (distinguished paper), ***PLDI 2021, PLDI 2017***
  - Released as OCaml 5.0
- **Impact**
  - 100+ million lines of OCaml, $27T USD per year in trading volume at Jane Street
  - Influenced React Hooks and WebAssembly Effect handlers (***OOPSLA 2023***)
  - 2023 SIGPLAN Programming Languages Software Award



Research Highlights (2)

Verified local-first software

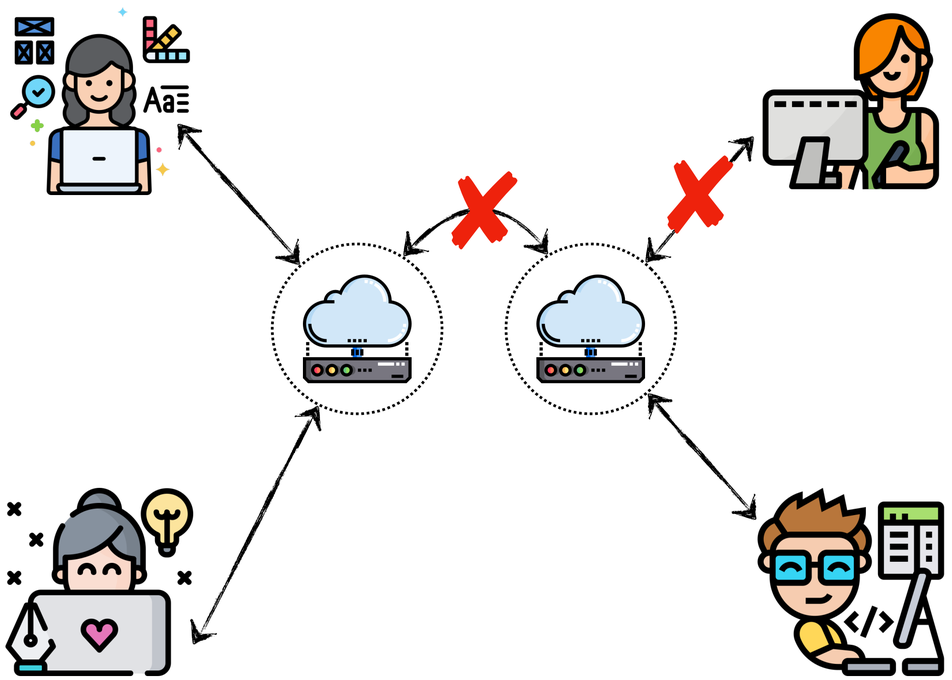
Airtable Google Docs Figma Notion Overleaf



Research Highlights (2)

Verified local-first software

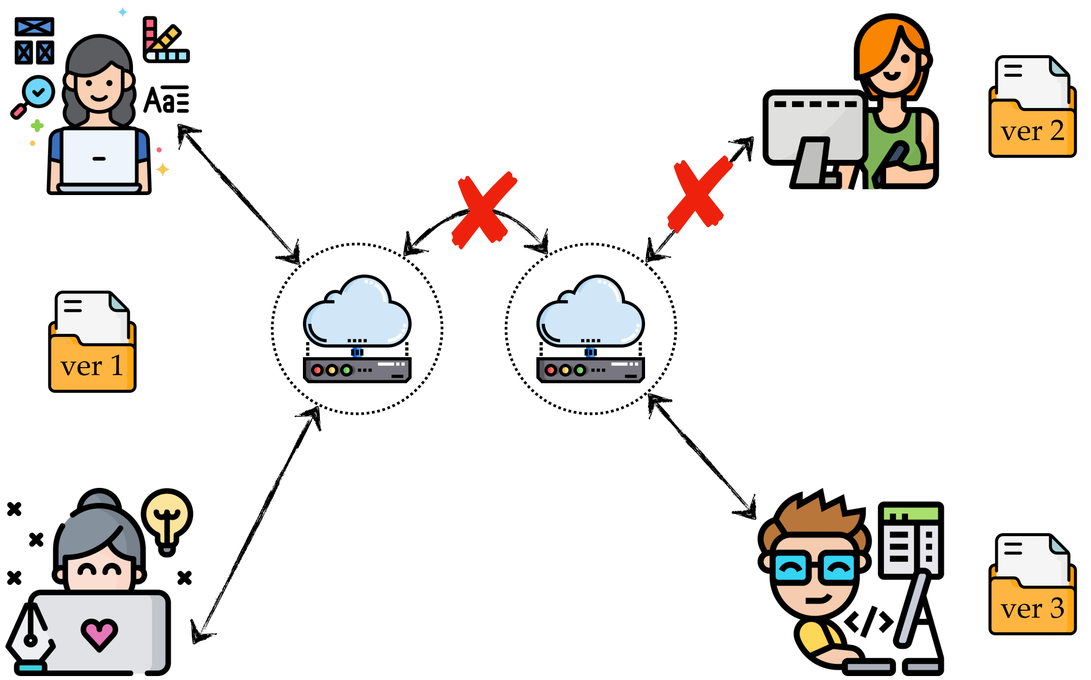
Airtable Google Docs Figma Notion Overleaf



Research Highlights (2)

Verified local-first software

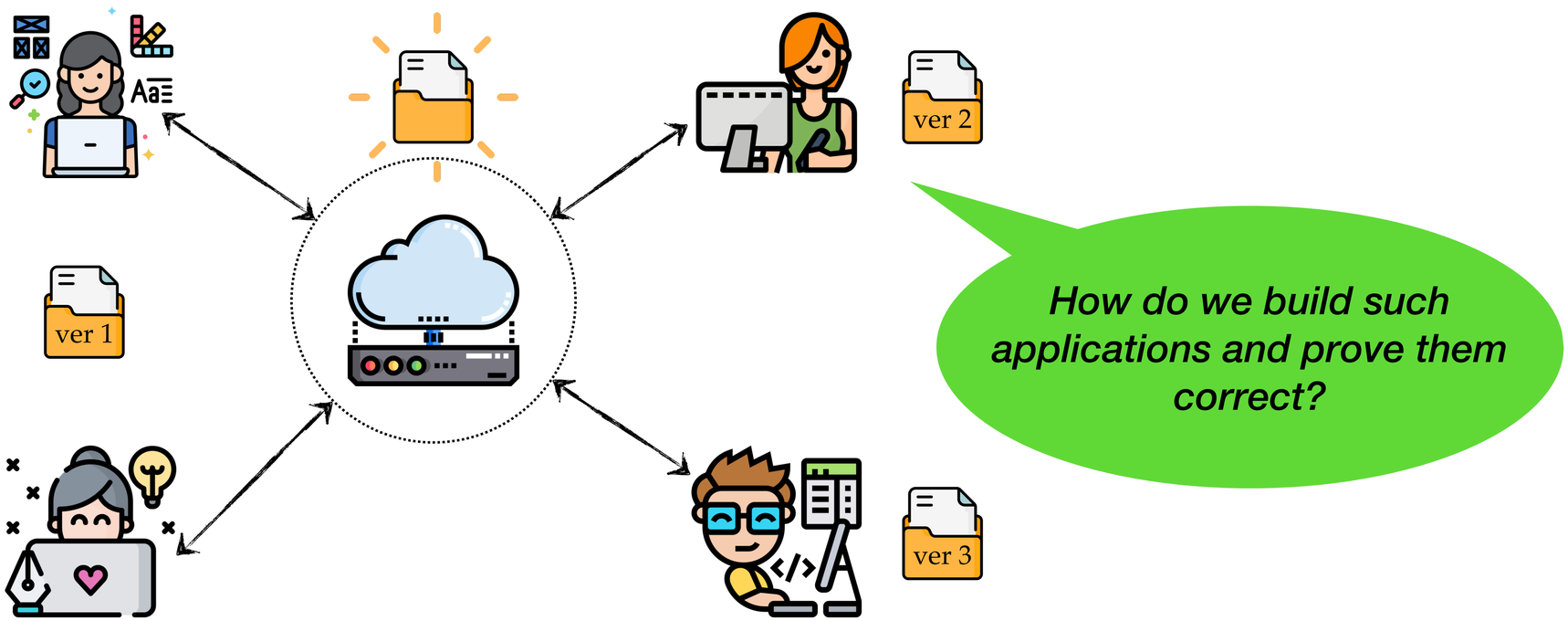
Airtable Google Docs Figma Notion Overleaf



Research Highlights (2)

Verified local-first software

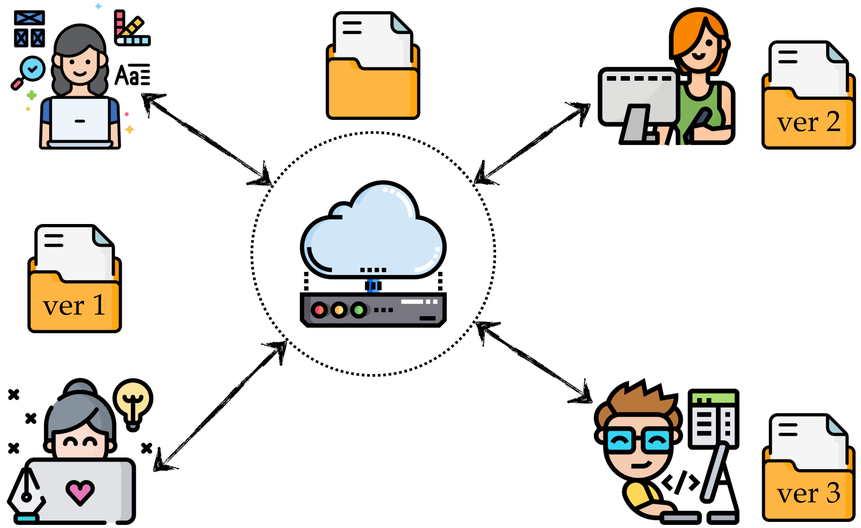
Airtable Google Docs Figma Notion Overleaf



Research Highlights (2)

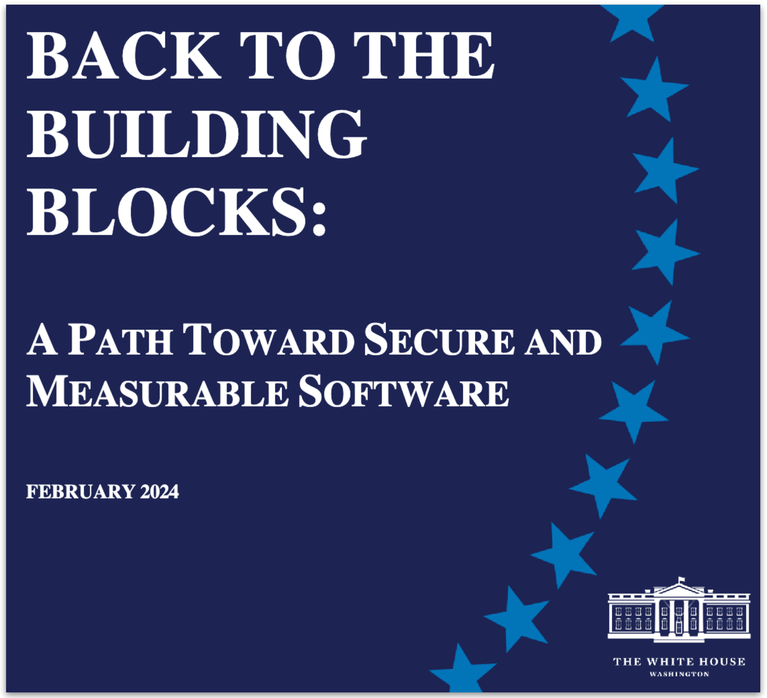
Verified local-first software

- Mergeable Replicated Data Types (MRDTs)
  - Replicated data types built on the principles of the ***Git*** DVCS
- Papers
  - Proof-oriented programming — ***PLDI 2022***, ***OOPSLA 2025***
  - Runtime system — ***APLAS 2020***



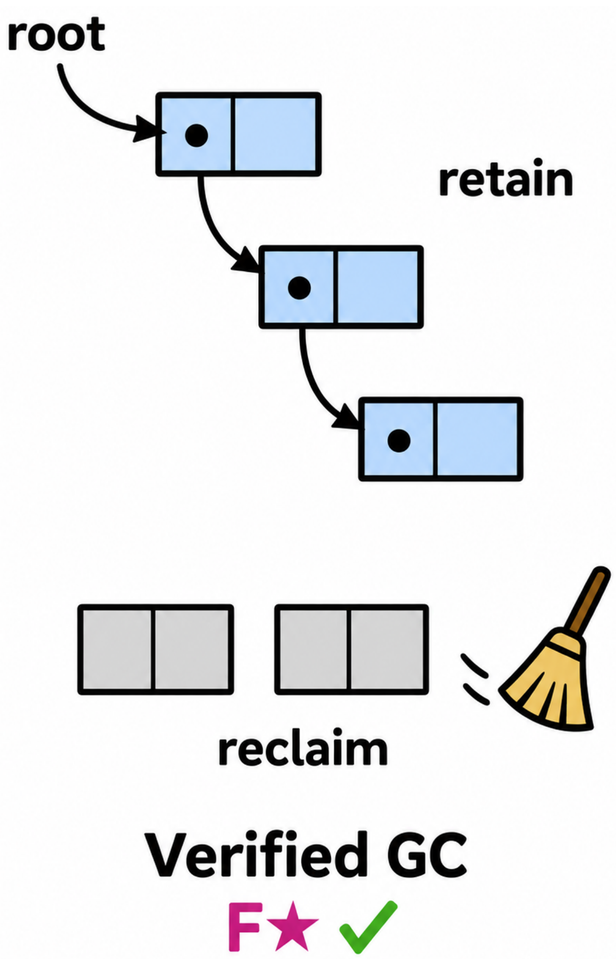
Research Highlights (3)

- **Securing the foundations**
  - The world runs on C/C++, but C/C++ is insecure
    - Memory unsafety $\Longrightarrow$ security vulnerability
  - Frontier models (e.g., Mythos) are extremely good at converting bugs to exploits
- **Multi-tiered defence**
  - Memory-safe languages ***(OCaml)***
  - Verification of low-level software
  - Secure hardware



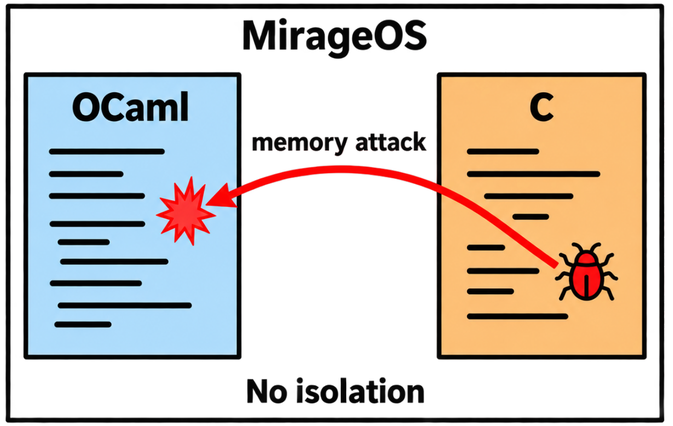
Research Highlights (3) — Securing the foundations

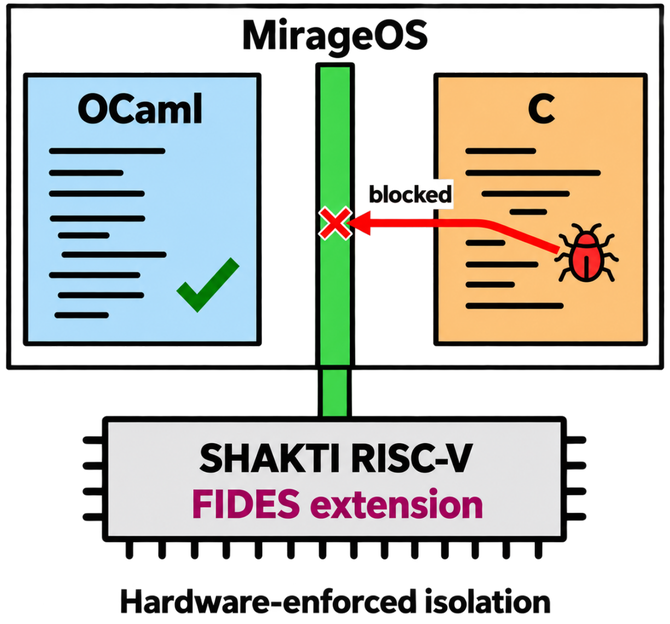
- OCaml is a type-safe language $\Longrightarrow$ memory-safe language
- ***OCaml GC written in C***
  - Can have bugs that invalidate OCaml's memory safety
- Mechanically verified GC (***JAR 2025***)
  - Provably correct GC in ***F****, a proof-oriented, SMT-aided programming language
- A simple mark-and-sweep GC, but designed to handle OCaml's memory model
  - Extract to C without losing performance; integrated with OCaml



Research Highlights (3) — Securing the foundations

- Unikernels — single-purpose OS+Application compiled to a bootable image
  - MirageOS — written in OCaml
  - **Advantages:** Smaller TCB, Lower resources
  - **Disadvantages:** C can violate OCaml guarantees, no isolation between independent modules
- FIDES: Compartments for Mixed Language Systems (***AsiaCCS 2026***)
  - Secure extension of the Shakti RISC-V processor for securing OCaml + C Unikernels
    - Hardware provides memory safety + compartments
    - OCaml code doesn't pay memory safety costs





FP Launchpad Centre (PI)



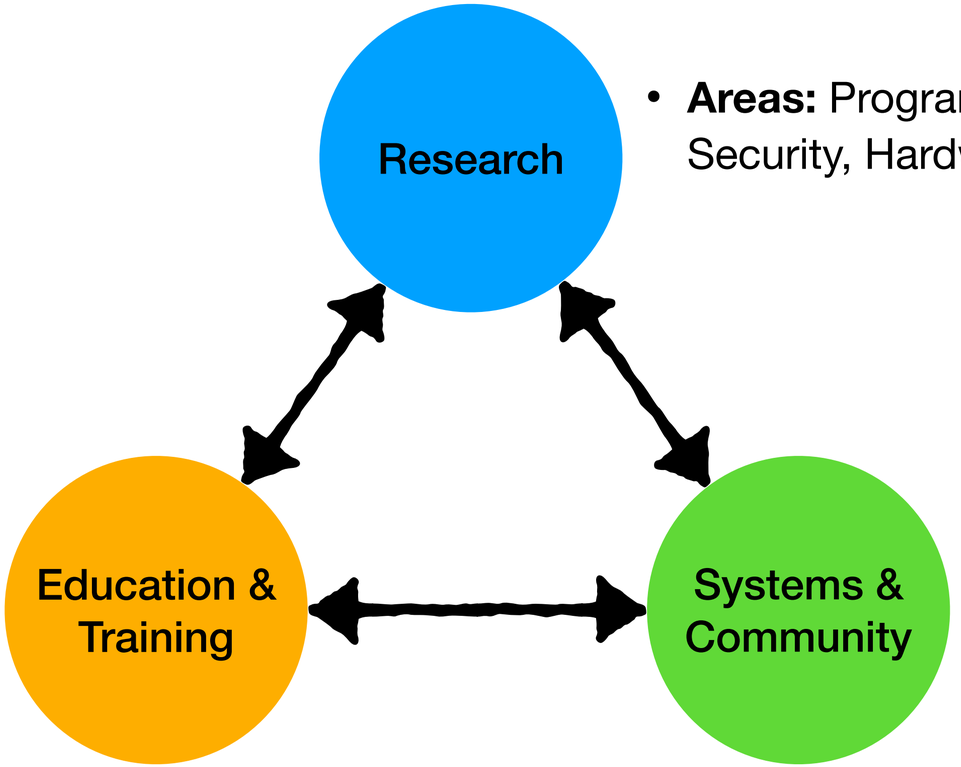
- A Centre for Functional Programming Research and Education
- Vision

  Build research and educational capacity to craft efficient, reliable, and trustworthy software with mathematical guardrails.

- Founding donor — Jane Street: **INR 23.42 crore** (USD 2.5M) over 4 years
- Institutionalises how my group already operates
  - Research hypotheses explored through deployed, open-source systems

Charter

- Post-bacc & doctoral fellowships
- Summer and winter schools, Dagstuhl-style research seminars
- Compiler hacking events
- Funds for visiting fellows + faculty



- **Areas:** Programming Languages, Security, Hardware Design, FM x AI

- Open source & industrial-strength
- **North Star projects**
  - Verifiable voting infrastructure
  - Digital public infrastructure for climate action
  - Robust foundational hardware & software stack

The bet

- AI agents write a growing share of the world's software
  - Human review cannot keep pace
- Answer → surround software with ***mathematical guardrails***
  - Languages, runtimes and verification tools that let machines check the properties we care about
- AI cuts the other way, too → Collapsing the cost of machine-checked proofs

Conjecture

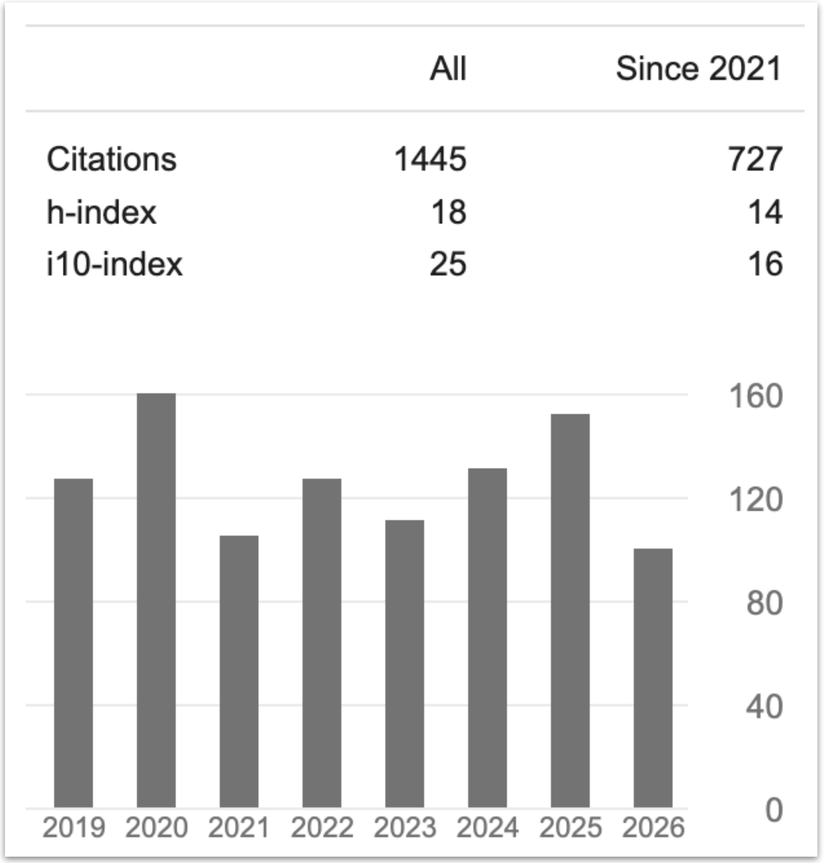
As humans step back from writing code, machines will target mathematically rigorous functional languages, as proof assistants already do.

A few bets are already paying off

- Microsoft Research collaborators extended the ***JAR 2025*** work with AI agents for complex ***verified*** GCs
  - A port and extension that would have taken person-years takes days!
  - https://github.com/FStarLang/pulse-verified-gc
- Agentic proof-oriented programming discovered previously unknown bugs in our ***OOPSLA 2025*** work
  - Co-developed a fix and extension, verified in Lean 4
  - https://github.com/fplaunchpad/sal
- **AI acceleration of broad scientific discovery and engineering feels imminent**
  - Formal verification keeps them honest

Quantitative Research Output

- 10 Journals, 18 Conference papers
  - 4 Journals and 5 conference papers at IITM
- 4 Distinguished paper awards
- Google Scholar metrics
  - 1416 citations, h-index 19, i10-index 26
- **Open-source software**
  - OCaml 5 is used widely in Industry and academia (https://ocaml.org)



Recognitions

- **ACM SIGPLAN Programming Languages Software Award 2023** (for OCaml)
- **Distinguished Paper Awards:** ICFP 2020, PADL 2021, MARC 2012, COORDINATION 2010.
- **Invited Keynotes:** ICFP 2022, ML Workshop 2022, PaPoC 2026, SERI 2026
- **Class of 1991 Award for Young Faculty:** IITM 2019
- **1851 Royal Commission** and **Darwin College, Cambridge** Research Fellowships
- **Maurice H. Halstead Award for Outstanding Research in Software Engineering**, Purdue 2014.

Research Scholar Guidance

- 2 PhD students graduated
  - Sheera Shamsu (2026) — **Role:** Primary Guide; **Co-guide:** Kartik Nagar
    - *“A Mechanically Verified Garbage Collector for OCaml”*
  - Vimala Soundarapandian (2026) — **Role:** Co-guide; **Primary Guide:** Kartik Nagar
    - *“Pragmatic Verification of Replicated Data Types”*
- 3 MS students graduated as sole guide
  - Shashank Shekhar Dubey (2021), Sumit Padhiyar (2021), Deepali Ande (2023)
- Ongoing
  - 3 PhD students — **Primary:** 1 (Durwasa); **Co-guide:** 2 (Sai, Nitish)
  - 2 MS students — **Sole guide:** 1 (Sanjeev); **Primary:** 1 (Vishakh)

Funded Projects

- INR 30.2 Cr in total funding
- **CSR & Sponsored Research**
  - FP Launchpad — Jane Street 23.42 Cr (PI)
  - Tezos Foundation — 1.79 Cr (PI)
- **Consultancy**
  - OCaml research strategy, technical leadership, upstreaming multicore, benchmarking — Tarides
  - Advising two IITM-alumni funded startups — ***Pramaana Labs*** and ***MemFold***
    - Compensation in equity through IC&SR; not counted in INR amount

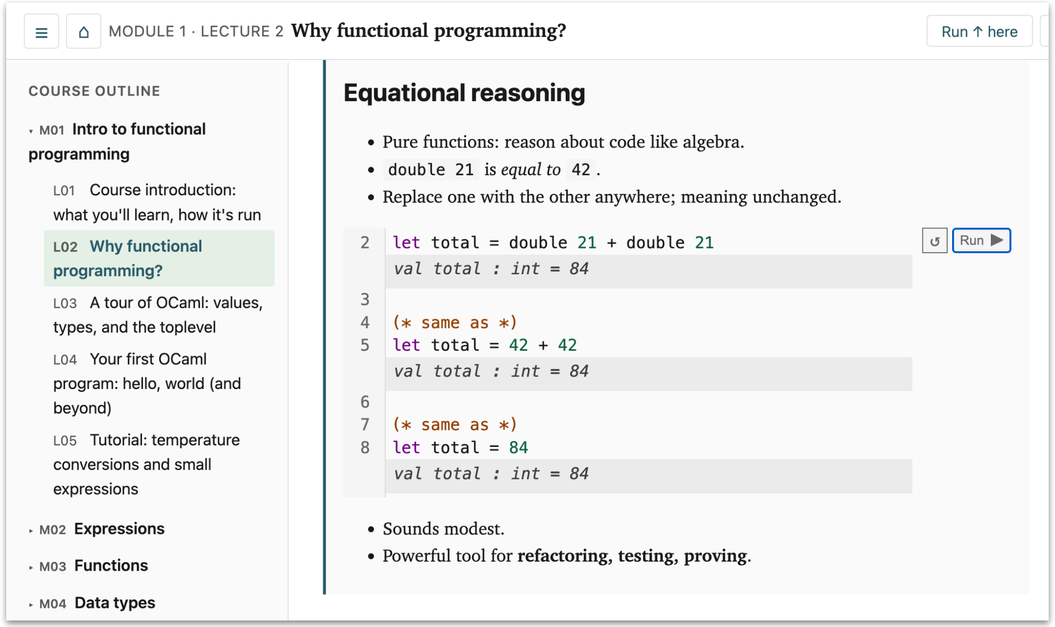
Teaching

- 5 core courses and 4 electives offered.
- Highlights
  - Introduced Programs and Proofs (CS6225); Revived Concurrent Programming (CS6868) after 8 years; redid Paradigms of Programming (CS3100) FP Part in OCaml (+ Jupyter)
- TCF scores are below

Course	Semester	Size	Course mean (dept)	Instr. mean (dept)
CS6868 Concurrent Programming	Jan–May 2026	51	**0.90** (0.79)	**0.92** (0.83)
CS3100 Paradigms of Programming	Jul–Nov 2025	108	0.74 (0.79)	0.79 (0.84)
CS6225 Programs and Proofs	Jan–May 2025	14	**0.92** (0.82)	**0.95** (0.88)
CS3300 Compiler Design	Jul–Nov 2022	103	0.81 (0.79)	0.85 (0.82)
CS6225 Programs and Proofs	Jan–May 2021	12	0.91 (0.84)	0.95 (0.85)
CS3100 Paradigms of Programming	Jul–Nov 2020	82	0.83 (0.82)	0.87 (0.85)
CS6225 Programs and Proofs	Jan–May 2020	11	0.86 (0.86)	0.93 (0.89)
CS3100 Paradigms of Programming	Jul–Nov 2019	74	0.73 (0.80)	0.80 (0.83)

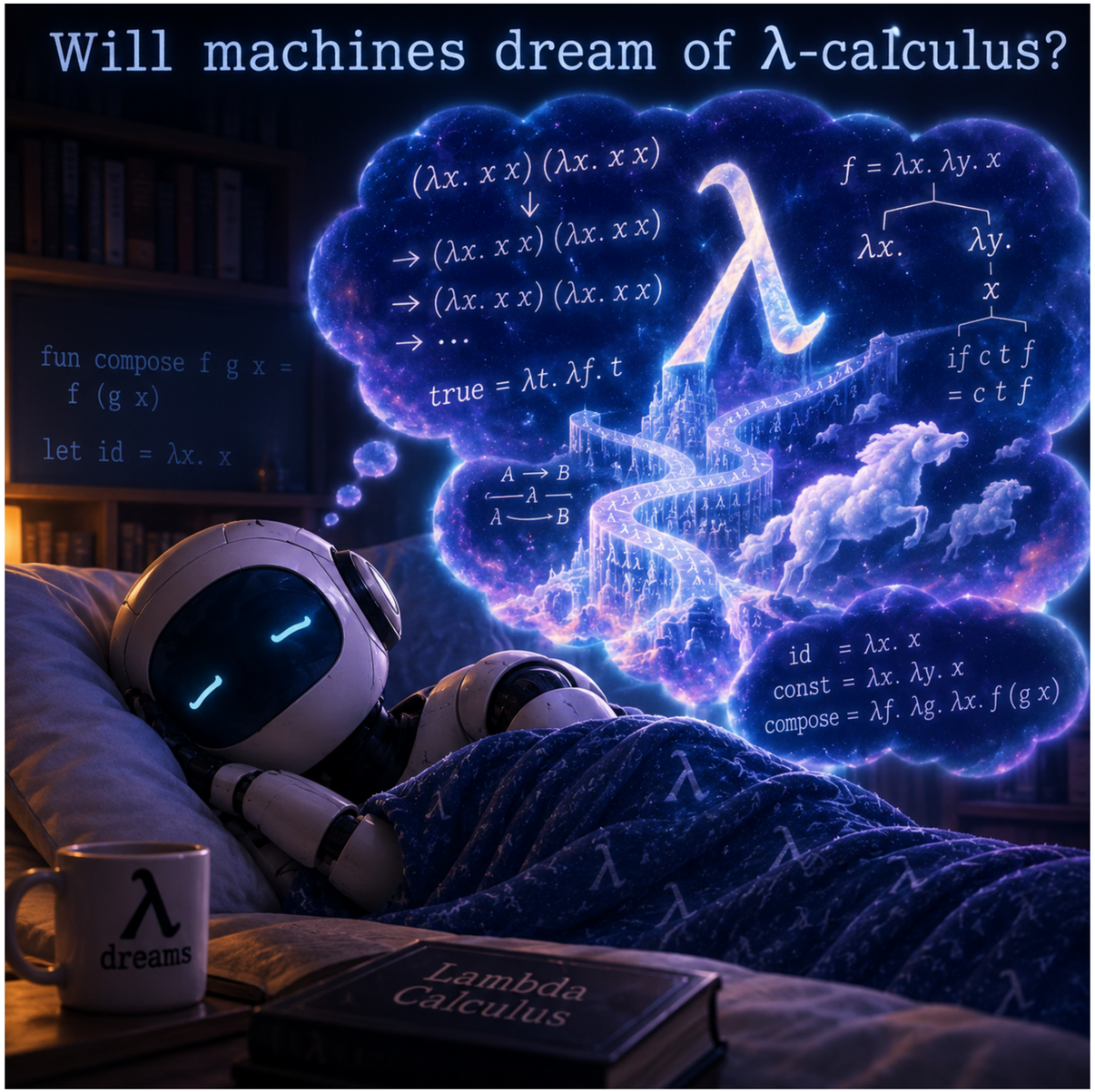
Teaching: Textbooks that run

- NPTEL Functional Programming in OCaml
  - Offering in Jul 2026 for the first time
  - 1227 enrolled users
  - Online, executable, fully client-side book
- Other teaching activities
  - Hands-on OCaml workshops at many conferences
  - Learning materials are quite popular on GitHub
    - OCaml effects handlers tutorial **724** 🌟; Effect handlers examples **456** 🌟; OCaml 5 parallel Programming Tutorial **300** 🌟



Service

- **Professional**
  - Core maintainer, OCaml (2021 — Present)
  - ICFP Steering Committee: Member-at-large and Diversity co-chair
  - SIGPLAN CARES Committee
  - Associate Editor: OOPSLA 2025
  - PC Chair: PROPL 2025, PaPoC 2022, ML Workshop 2019; AEC Chair, ESOP 2022
  - 30+ Program Committees incl POPL, PLDI, ICFP, OOPSLA
- **Institute and Department**
  - Board member, IC&SR, IIT Madras (2021–2023)
  - Technical Committee, Dept. of School Education, Govt. of Tamil Nadu (2020–2022)
  - Department-level activities incl. department vision, website redesign, facad, MS/PhD QP Setting, etc.


Will machines dream of λ-calculus?
(λx. x x) (λx. x x)
↓
→ (λx. x x) (λx. x x)
→ (λx. x x) (λx. x x)
→ ⋯
true = λt. λf. t
f = λx. λy. x
λx.
λy.
x
if c t f
= c t f
fun compose f g x =
f (g x)
let id = λx. x
A → B
A
A B
id = λx. x
const = λx. λy. x
compose = λf. λg. λx. f (g x)
λ
dreams
Lambda
Calculus

Thank you!