Když pravá ruka neví, co dělá ta levá

JOpenSpace 2025, Telč 3.-5.10.

Pavel Lahoda, Actiwerks

V minulém díle jste viděli…

Distributed serverless architecture

JOpenSpace 2024, Telč 4.-6.10.

Pavel Lahoda, Actiwerks

Internet Computer (DFinity)

https://dfinity.org

Tenhle příběh pokračuje dál…

Caffeine

- „Self writing internet“
- Runs on IC (Motoko)
- React with TypeScript, TanStack Query, Tailwind CSS

like help with anything e...
Yesterday 11:42

🚀 Going live with draft Version 11, now deploying to production... 11:55

🎉 Draft Version 11 is now LIVE in production!

Live app web page

11:55

⏰ Your draft app has expired. Do you want to rebuild it? 13:42

Today

Ok, building your code, and deploying to the network 🍿 5m ago

🎬 Congrats, your first app draft Version 11 is ready for review

Draft app web page

Actions

4m ago

Can you confirm that the new scrolling, media upload support, and quick post features are working smoothly as you envisioned? Would you like assistance with anything else? 4m ago

Enter instructions or question

SociaLies

Overview

SociaLies is a modern social media platform focused on microblogging and community interaction with Internet Identity authentication. Users can create posts with text, images, or short videos, engage through comments and likes, build social connections via following relationships, and manage their profiles. The platform features a clean, minimalistic design with a cosmic-themed color palette and responsive layout optimized for both desktop and mobile devices. Posts can be shared externally via unique URLs for focused viewing.

Authentication

- Users must authenticate using Internet Identity before accessing platform features
- Anonymous users can view public content but cannot create posts or interact with content
- Profile setup is required after first authentication with username validation
- Login and logout functionality integrated throughout the application interface

Core Features

User Profile Management

- Comprehensive user profiles with username, display name, bio, and profile picture asset keys
- Profile creation requires unique username validation with real-time availability checking
- Users can update their profile information including display name, bio, and profile picture

SL SociaLies Login

Pavel
1 day ago

1 1 Share

Pavel
1 day ago

Kandidát na premiéra Andrej Babiš oznámil, že po volbách bude usilovat o přijetí referenda o své eutanázii. Tomu se říká sebereflexe!

0 0 Share

SL SociaLies

https://socialies-nfw.caffeine.xyz/

Private Compute

- Apple Private Compute
- Microsoft Azure Confidential Computing
- Amazon

TEE: Based on trust (Backdoor possible)

Compute directly on encrypted data — without ever decrypting it

Homomorphic Encryption

Homomorphic *(adj.)*: from Greek *homo-* (“same”) + *-morph* (“form” or “structure”).

In mathematics: a homomorphism is a structure-preserving map between two algebraic systems.

Enc(a) ⊕ Enc(b) = Enc(a ⊗ b)

(Encrypted operation yields encrypted result)

- You can add, multiply, or run entire programs on encrypted data.
- The server never sees raw inputs or outputs — it just manipulates ciphertext.
- When decrypted, the result is mathematically correct, as if you ran the computation directly on the original values.

$$Enc(m) = m + 2r + pq$$

- m = message (e.g. 5)
- r = small random noise
- q = large public modulus
- p = secret key (large prime)

Microsoft SEAL

- C++
- Full HE support (CKKS, BFV schemes)

Zama Concrete (tfhe-rs)

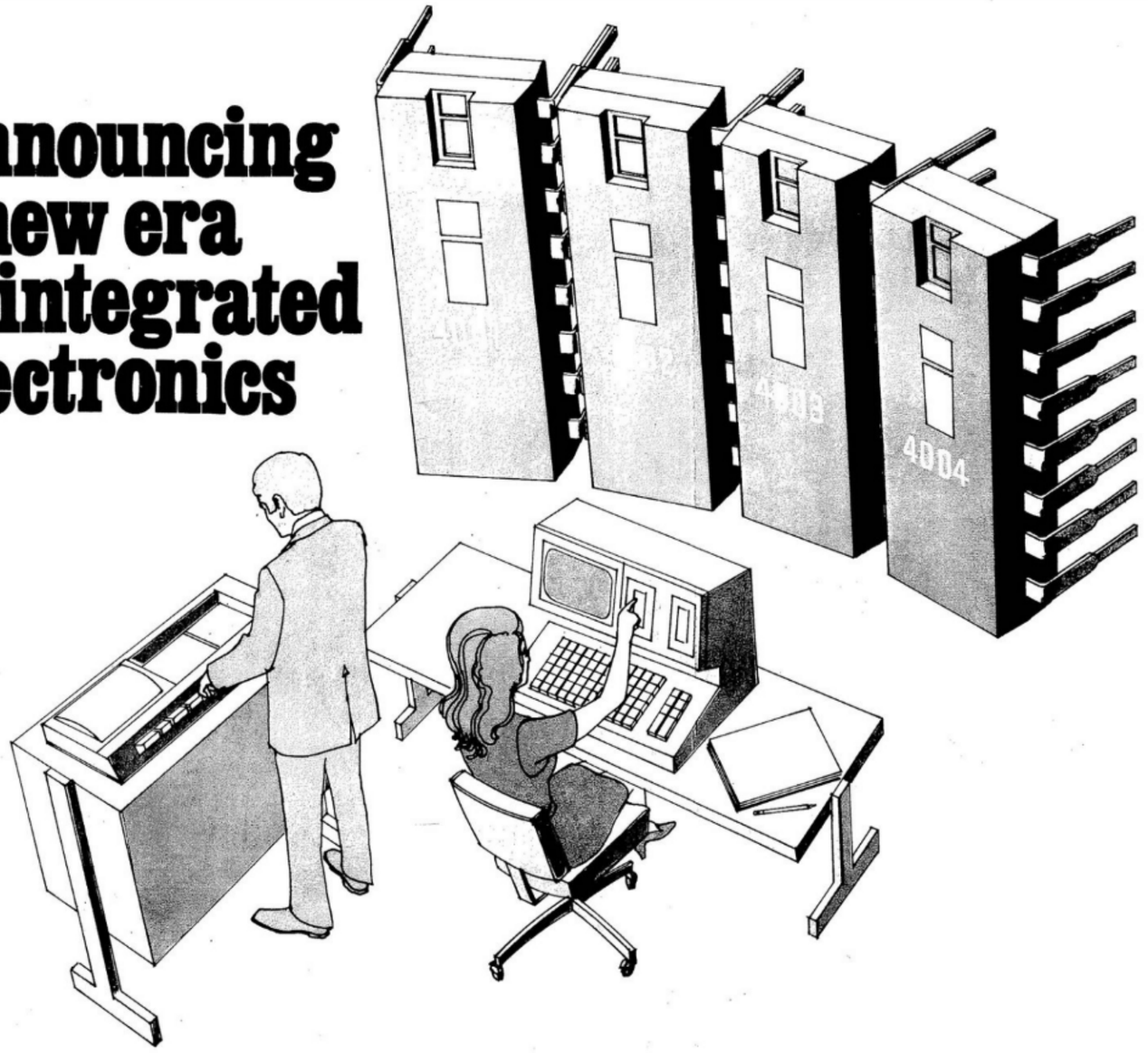
- Rust
- TFHE (boolean circuits, short bootstraps)

Canonical example

```rust
use tfhe::prelude::*;

let (client_key, server_key) = tfhe::generate_keys();

// Encrypt two bits
let ct_a = client_key.encrypt(true);
let ct_b = client_key.encrypt(false);

// Homomorphic AND
let ct_result = server_key.and(&ct_a, &ct_b);
let decrypted = client_key.decrypt(&ct_result);

assert_eq!(decrypted, false);
```

Usage

Announcing a new era of integrated electronics



A micro-programmable computer on a chip!

Intel introduces an integrated CPU complete with a 4-bit parallel adder, sixteen 4-bit registers, an accumulator and a push-down stack on one chip. It's one of a family of four new ICs which comprise the MCS-4 micro computer system—the first system to bring you the power and flexibility of a dedicated general-purpose computer at low cost in as few as two dual in-line packages.

MCS-4 systems provide complete computing and control functions for test systems, data terminals, billing machines, measuring systems, numeric control systems and process control systems.

The heart of any MCS-4 system is a Type 4004 CPU, which includes a powerful set of 45 instructions. Adding one or more Type 4001 ROMs for program storage and data tables gives you a fully functioning micro-programmed computer. To this you may add Type 4002 RAMs for read-write memory and Type 4003 registers to expand the output ports.

Using no circuitry other than ICs from this family of four, you can create a system with 4096 8-bit bytes of ROM storage and 5120 bits of RAM storage. When you require rapid turn-around or need only a few systems, Intel's erasable and re-programmable ROM, Type 1701, may be substituted for the Type 4001 mask-programmed ROM.

MCS-4 systems interface easily with switches, keyboards, displays, teletypewriters, printers, readers, A-D converters and other popular peripherals.

The MCS-4 family is now in stock at Intel's Santa Clara headquarters and at our marketing headquarters in Europe and Japan. In the U.S., contact your local Intel representative for technical information and literature. In Europe, contact Intel at Avenue Louise 216, B 1050 Bruxelles, Belgium. Phone 492003. In Japan, contact Intel Japan, Inc., Parkside Flat Bldg. No. 4-2-2, Sendagaya, Shibuya-Ku, Tokyo 151. Phone 03-403-4747.

Intel Corporation now produces micro computers, memory devices and memory systems at 3065 Bowers Avenue, Santa Clara, Calif. 95051. Phone (408) 246-7501.

intel® delivers.

Homomorphic FPGA Intel 4004

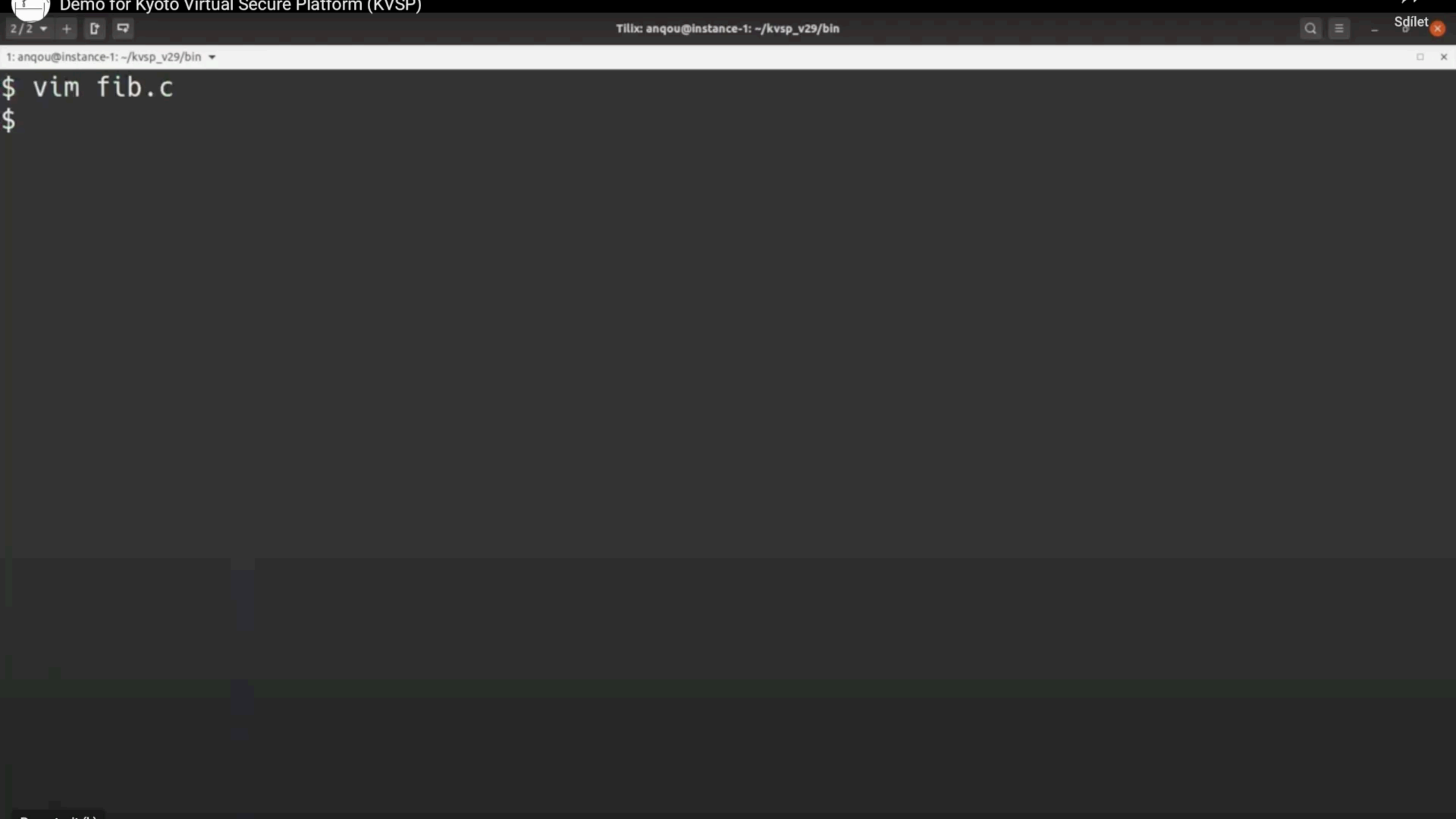
https://www.zama.ai/post/homomorphic-fpga-implementation-of-the-intel-4004-part-1

Virtual Secure Platform

Kyoto (KVSP) https://virtualsecureplatform.github.io/

Demo

```c
static int fib(int n) {
  int a = 0, b = 1;
  for (int i = 0; i < n; i++) {
    int tmp = a + b;
    a = b;
    b = tmp;
  }
  return a;
}

int main(int argc, char **argv) {
  // Calculate n-th Fibonacci number.
  // n is a 1-digit number and given as command-line argument.
  return fib(argv[1][0] - '0');
}
```


Demo for Kyoto Virtual Secure Platform (KVSP)
2/2
Tilix: anqou@instance-1: ~/kvsp_v29/bin
Sdílet
1: anqou@instance-1: ~/kvsp_v29/bin
$ vim fib.c
$

Run your own

AWS EC2 (XLarge, Metal)

CPU with AVX2 support (e.g. Intel Core i7-8700)
16GB RAM
NVIDIA GPU (not required but highly recommended)
Only NVIDIA V100 and A100 are supported.
Other GPUs may work but are not supported.

ZAMA Protocol

Zama Confidential Blockchain Protocol: HTTPZ

https://docs.zama.ai/protocol/zama-protocol-litepaper

Q & A

0553A01AC2765DF74683405FCC8CF44554108DBF8C937A3E5F2B73EEAB7C57DB1D

@LAHODA.BSKY.SOCIAL

HTTPS://GITHUB.COM/ACTIWERKS