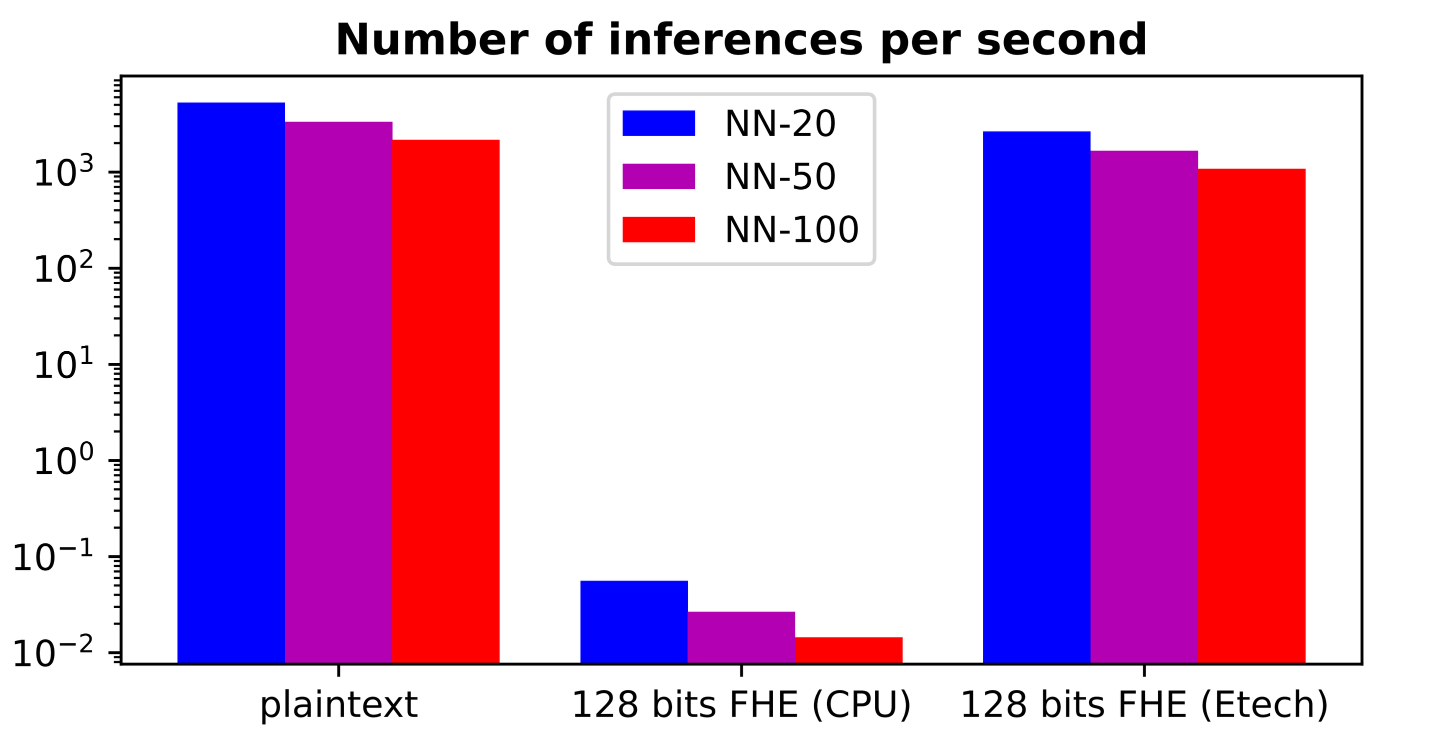
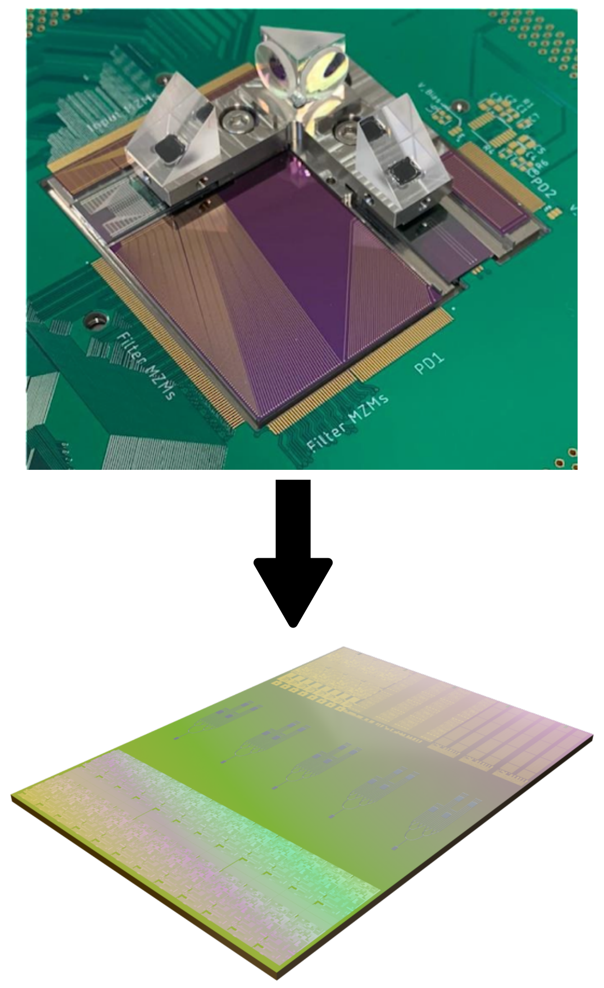
CONCRETE AND SPECIALIZED ACCELERATORS

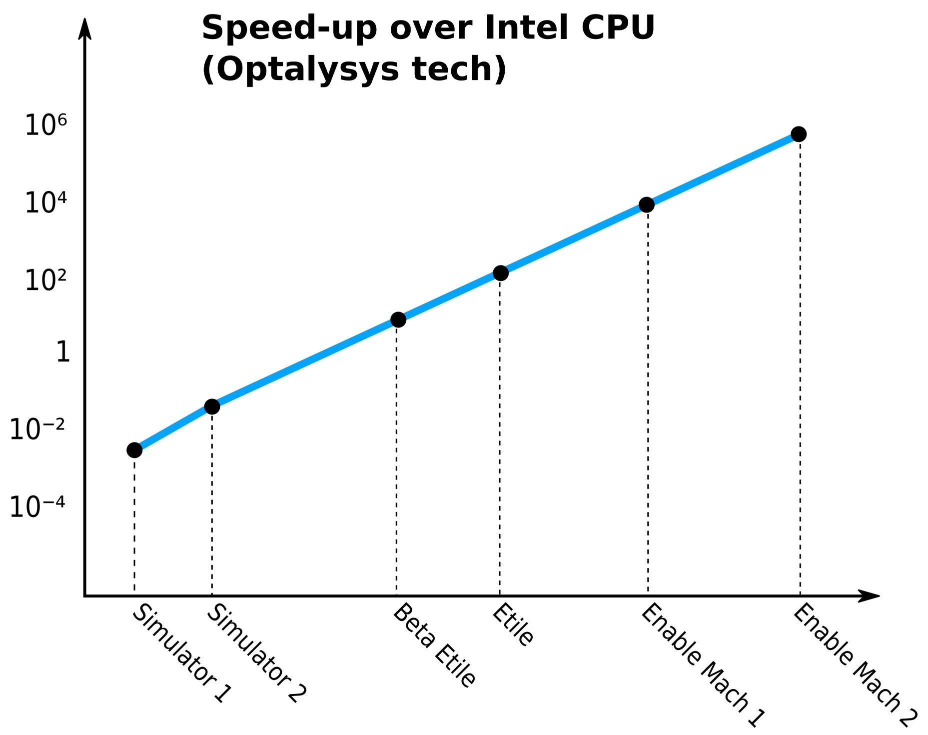
FAST

- The Number of AI inferences per second of an MNIST fully-connected neural network with 92 neurons per hidden layer is a **factor 10^5 slower** when evaluated homomorphically on a CPU https://whitepaper.zama.ai



- The Concrete Optalysys backend provides tools to **benchmark the WIP software stack** and **estimate the performance of upcoming hardware**, to optimize the software interface and high-level algorithms, and to determine the requirements for the host hardware



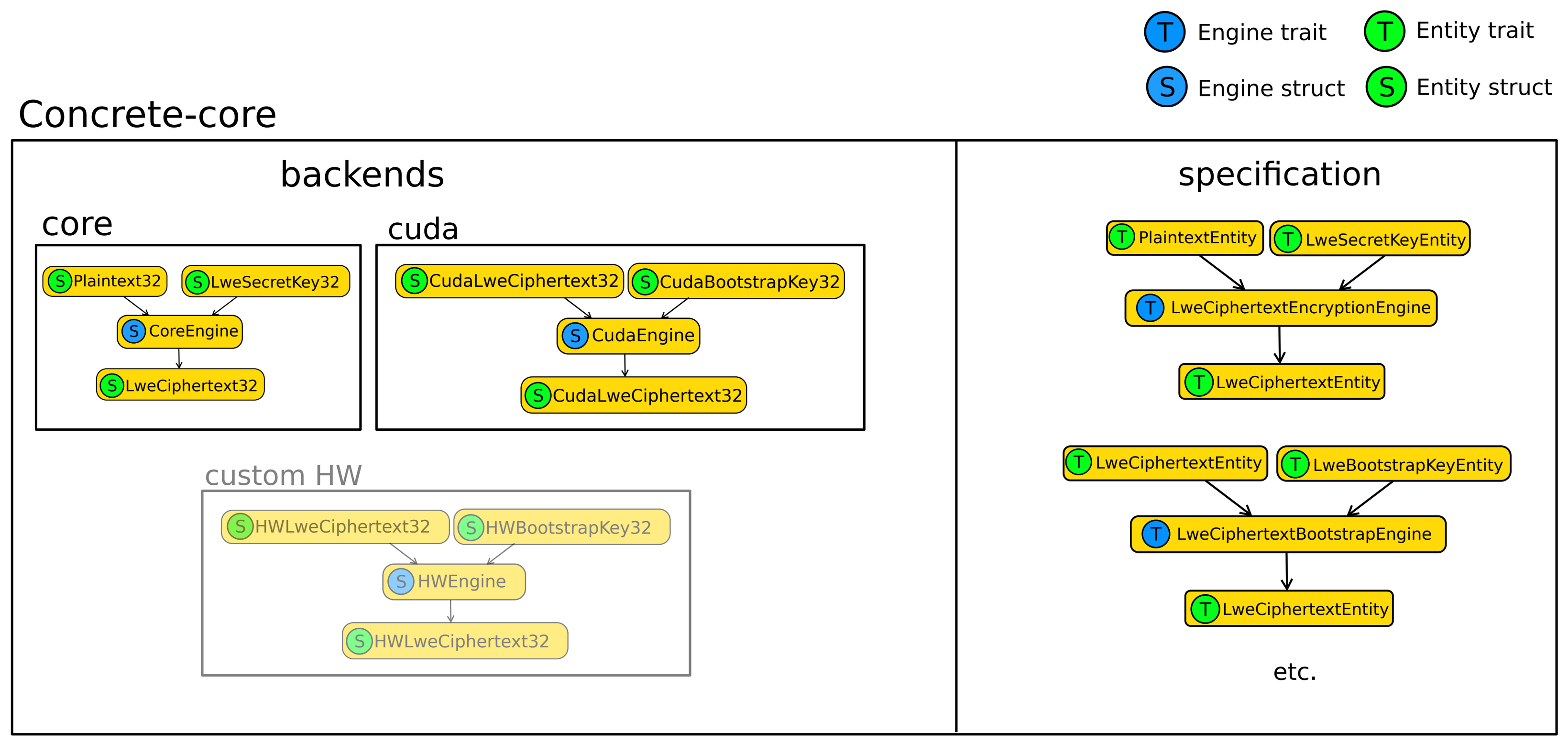


From table-top lab system to SiPh chip integration

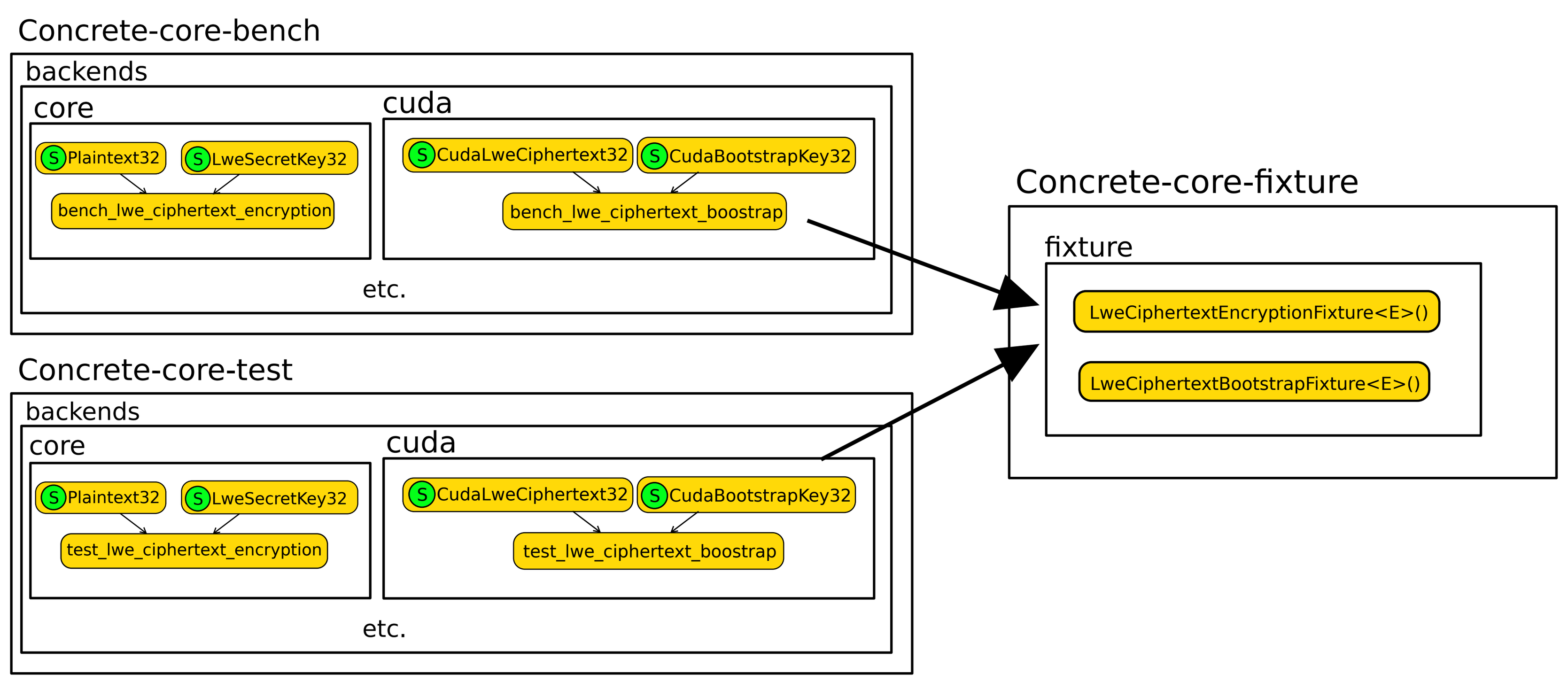
Towards 10^6 x FHE acceleration

EASY

- `concrete-core` Entities (Datatypes) and Engines (Functions) can be "overloaded" with the API of a new accelerator
- Integration of the Optalysys backend in **two workdays**!



- `concrete-core` has fixtures to **easily test and benchmark new accelerators**
- Setting up tests and benches for the Optalysys backend in **half a workday**!

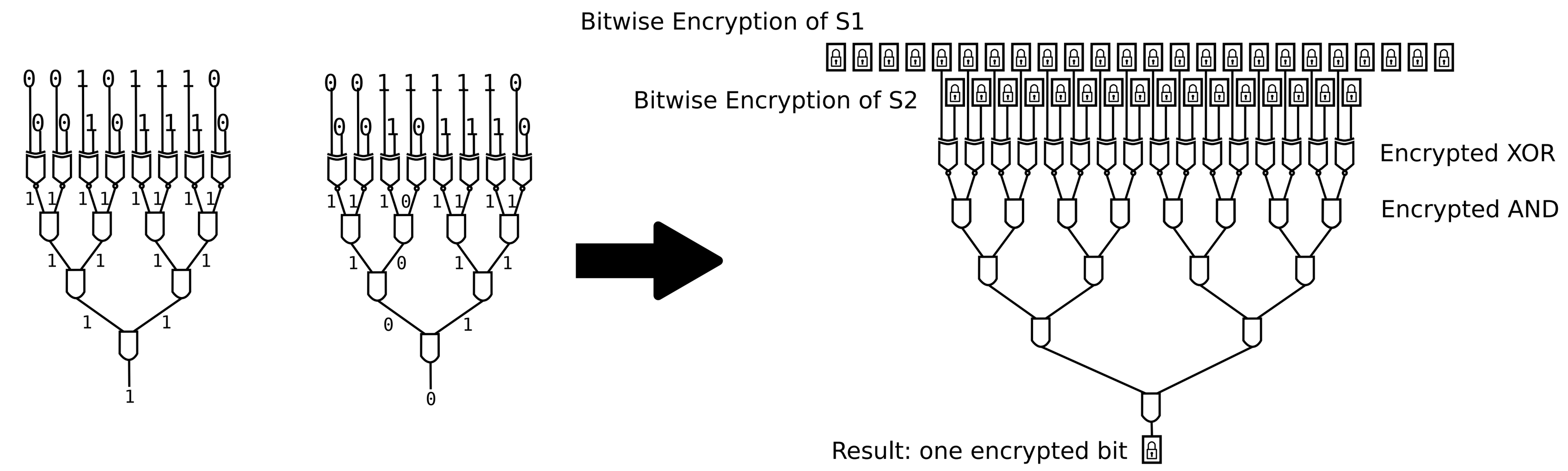


CONCRETE FHE Library

https://docs.zama.ai

ACCESSIBLE

- Strings can be compared character by character using a Boolean circuit
- The resulting circuit can be converted to a homomorphic program using `concrete-boolean`



When comparing 8-bit characters with Boolean circuits, the result is 1 when the characters are equal and 0 when they are unequal

- Coding Homomorphic String Search (and other functions) **requires no knowledge of cryptography**!
- More info at https://optalysys.com/encrypted-search-using-fully-homomorphic-encryption

```rust
/// Return a ciphertext that decrypts to `true` if `a` and `b` encrypt the same bit and `false`
/// otherwise.
pub fn bits_are_equal(server_key: &ServerKey, a: &Ciphertext, b: &Ciphertext)
    -> Ciphertext
{
    server_key.xnor(&a, &b)
}


/// If `a` is not empty and `a` and `b` have the same length, return `Ok(c)` where `c` is ciphertext
/// that decrypts to `true` if `a` and `b` encrypt the same sequence of bits and `false`otherwise.
/// Return an `FHEError` if `a` is empty or if `a` and `b` have different lengths.
pub fn are_equal(server_key: &ServerKey, a: &[Ciphertext], b: &[Ciphertext])
    -> Result<Ciphertext, FHEError>
{

    // check that a is not empty
    if a.len() == 0 {
        return Err(FHEError::new(
            "Error checking the equality between two elements: the first element is empty"
            .to_string(),
        ));
    }

    // check that the two inputs have the same size
    if a.len() != b.len() {
        return Err(FHEError::new(format!(
            "Error checking the equality between two elements: the elements have different lengths ({} and {})",
            a.len(), b.len()
        )));
    }

    // check the equality of the first elements
    let mut are_equal = bits_are_equal(server_key, &a[0], &b[0]);

    // check the equality of the other elements
    for i in 1..a.len() {
        are_equal = server_key.and(&are_equal, &bits_are_equal(server_key, &a[i], &b[i]));
    }

    Ok(are_equal)
}
```

See our websites for more examples

ZAMA

FLORENT MICHEL (OPTALYSYS) // THOMAS DE CNUDDE (ZAMA) // AGNES LEROY (ZAMA)
with contributions from JOSEPH WILSON (OPTALYSYS) // ALEX PERE (ZAMA)

// **FHE.ORG CONFERENCE // MAY 2022**