

# **Futures und Promises**

**Umsetzung in verschiedenen Programmiersprachen**

# Futures und Promises

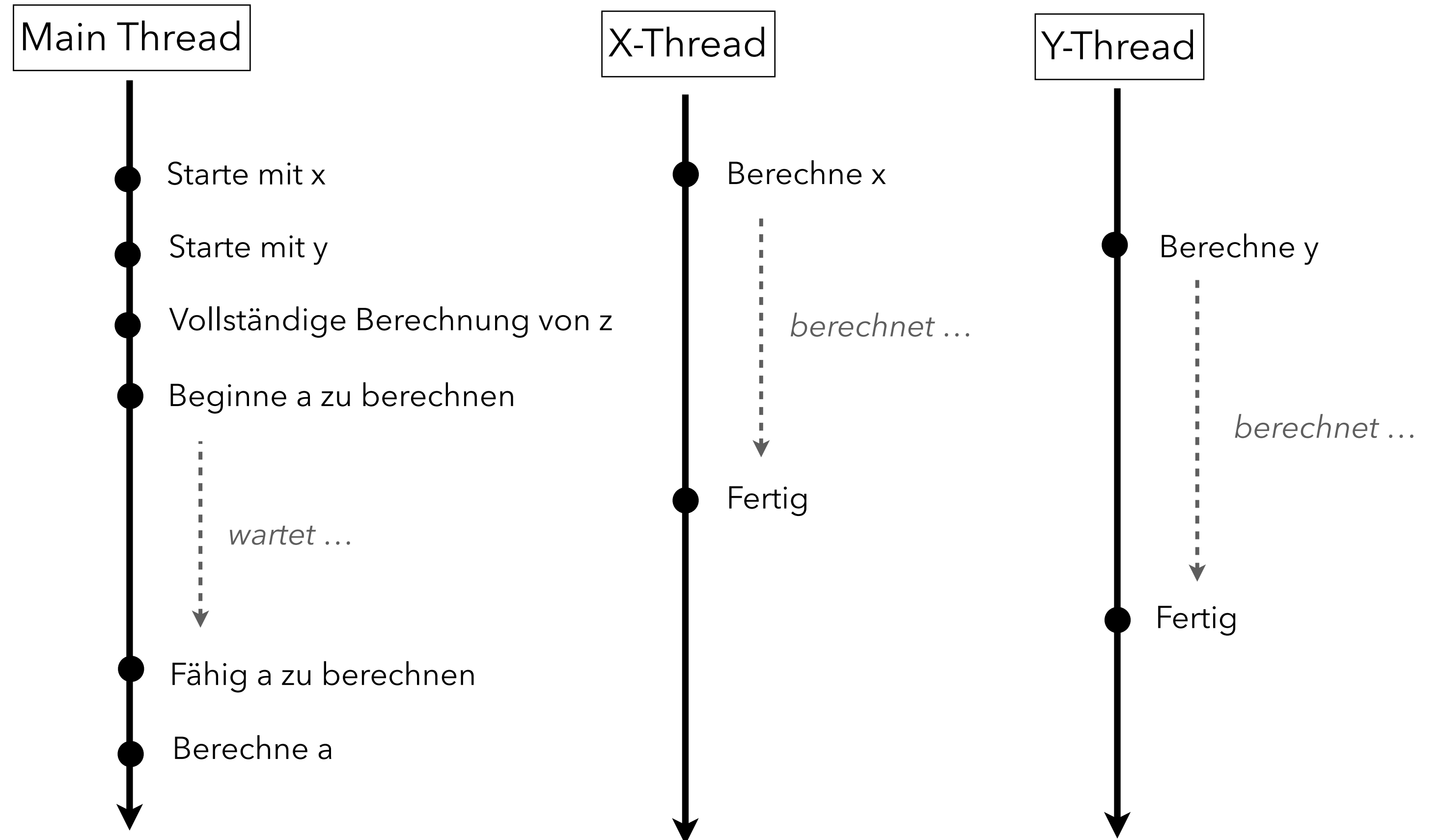
## Definition

- *High Level Concurrency Konstrukt zur Unterstützung asynchroner Programmierung (~Scala)*
- Das Future ist ein Platzhalter für eine Berechnung, dessen Ergebnis noch nicht existiert
  - Wert wird parallel bereitgestellt um anschließend verwendet zu werden.
- Promise ist ein Versprechen das jenes Ergebnis eintreten wird

# Futures und Promises

## Beispiel

```
x @= calc_x()  
y @= calc_y()  
z := calc_z()  
  
a := x + y + z
```



# Futures und Promises

C++

HL  
KA  
A

```
// ...

void add(std::promise<int>&& promise, int a, int b) {
    promise.set_value(a+b);
}

int main() {
    // ...

    std::promise<int> promise;
    std::future<int> future = promise.get_future();
    std::thread thread(add, std::move(promise), a, b);
    int result = future.get();
    thread.join();

    // ...
}
```

# Futures und Promises

## Go

```
// ...
```

```
func add(a int, b int) Future {  
    return future(func() (int) {  
        return a + b  
    })  
}
```

```
func future(function func() (int)) Future {  
    channel := make(chan futureType)  
    go func() {  
        value := function()  
        fType := futureType{value}  
        for {  
            channel <- fType  
        }  
    }()  
    return channel  
}
```

# Dankeschön

## Fragen?

Raphaele Licciardo - 60559