FORECASTING TIME-SERIES WITH POLARS AND DENO

Piotr Stepinski
stepinski

- from se to data science
- epic stuff with time-series @ infinitii ai make your data smarter
- father of 6 and husband to 1, beginner rope jumper

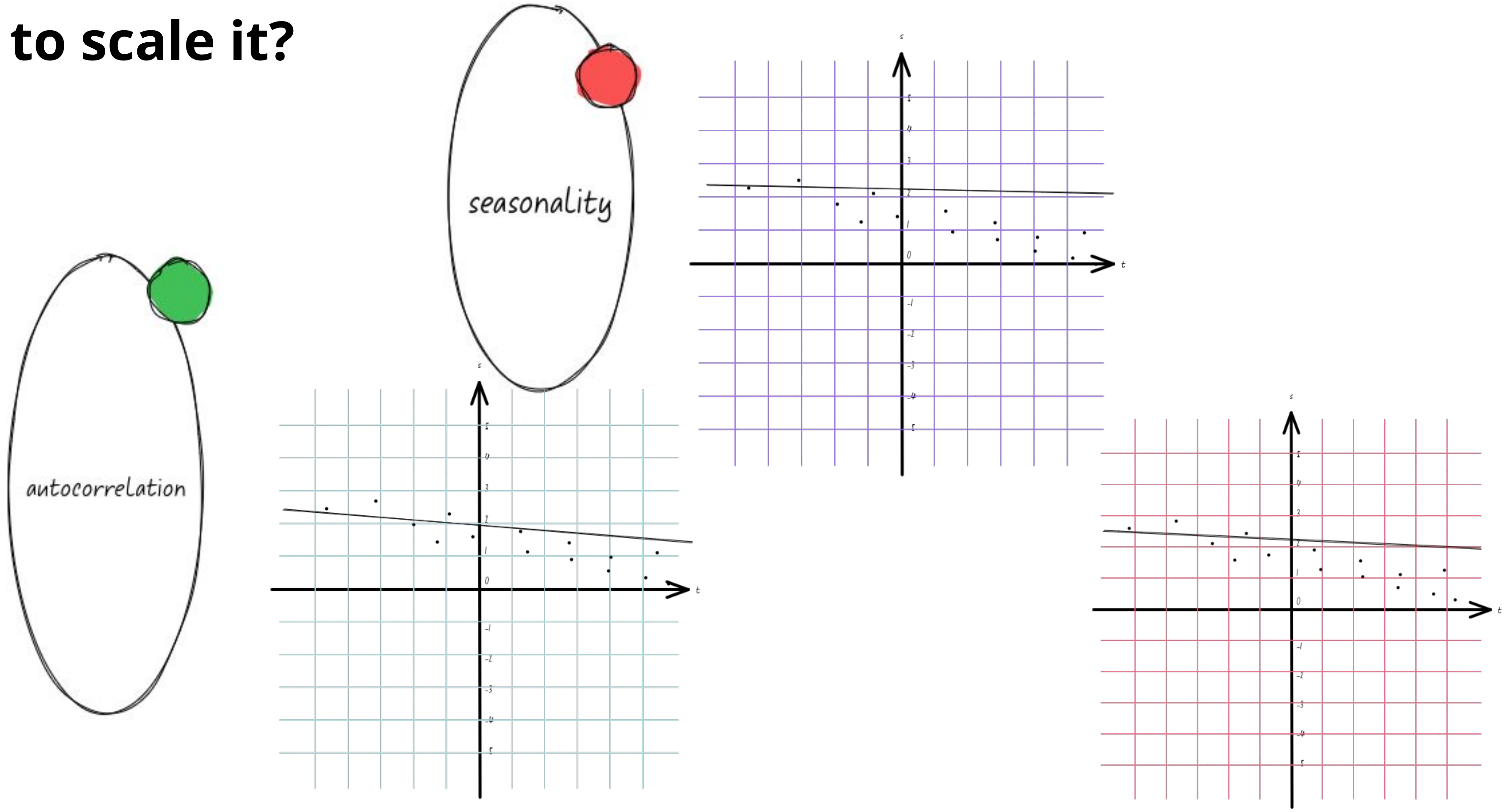
key takeaways

- the story
- torment of wasm
- building daily patterns
- deus ex machina

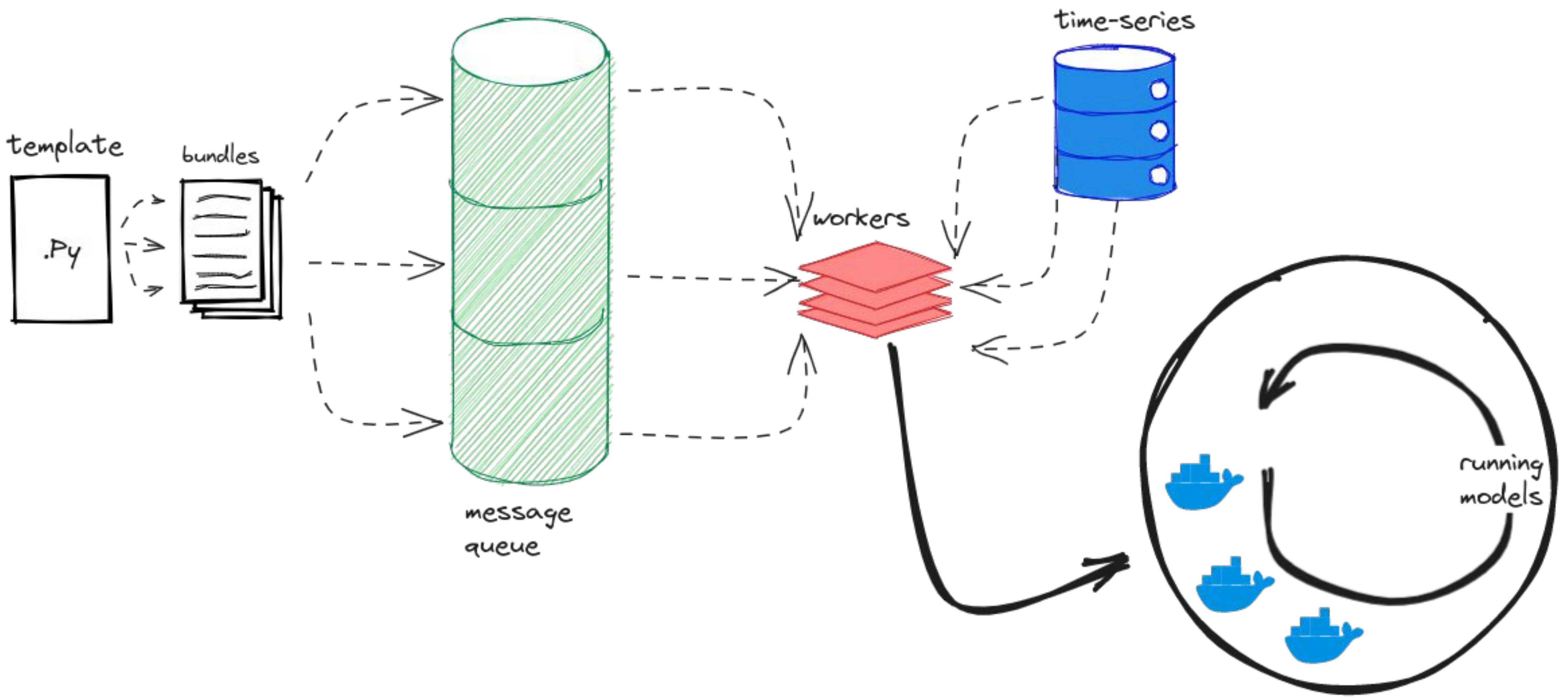
the story

- city of York
- water level sensors
- outlier detection

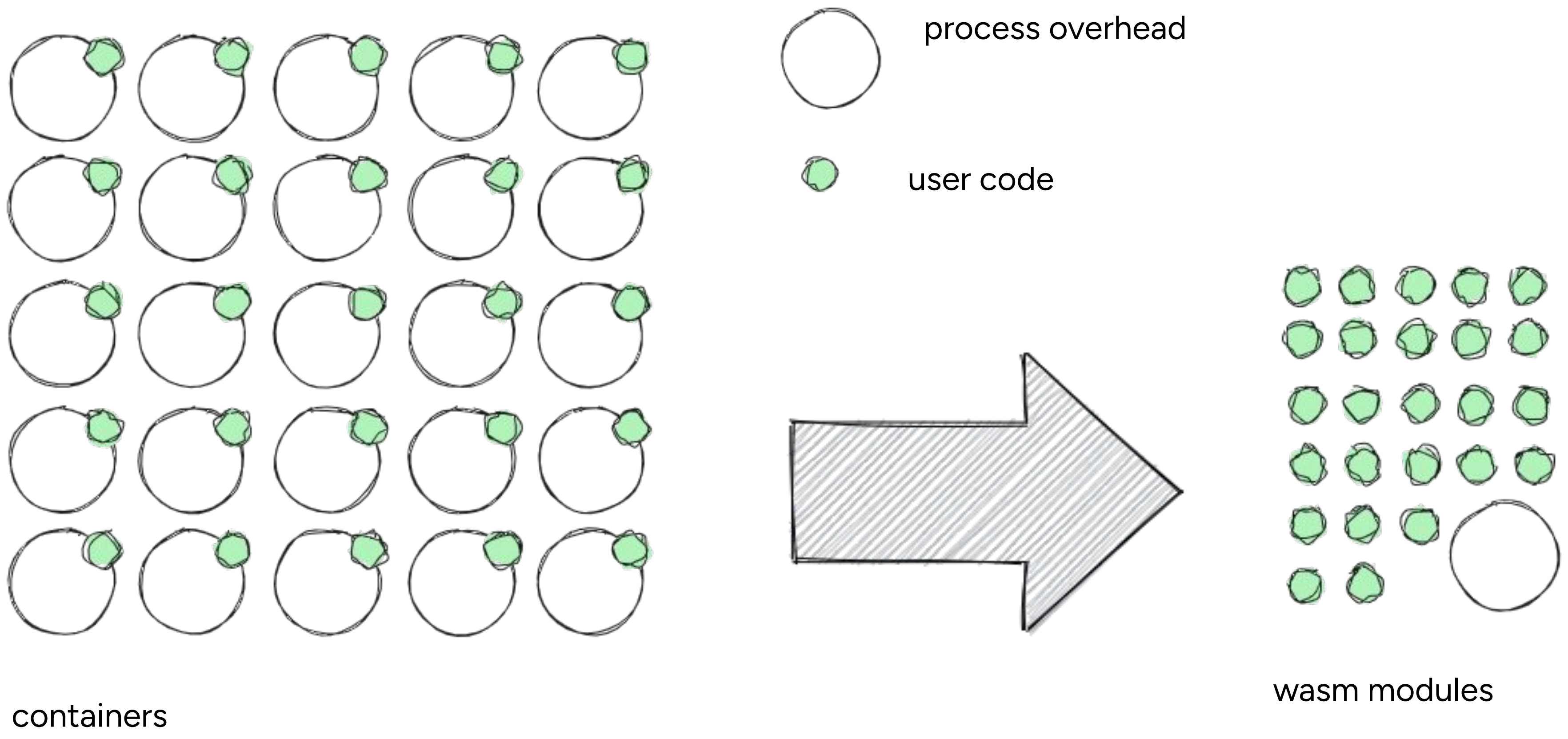
how to scale it?



pipeline

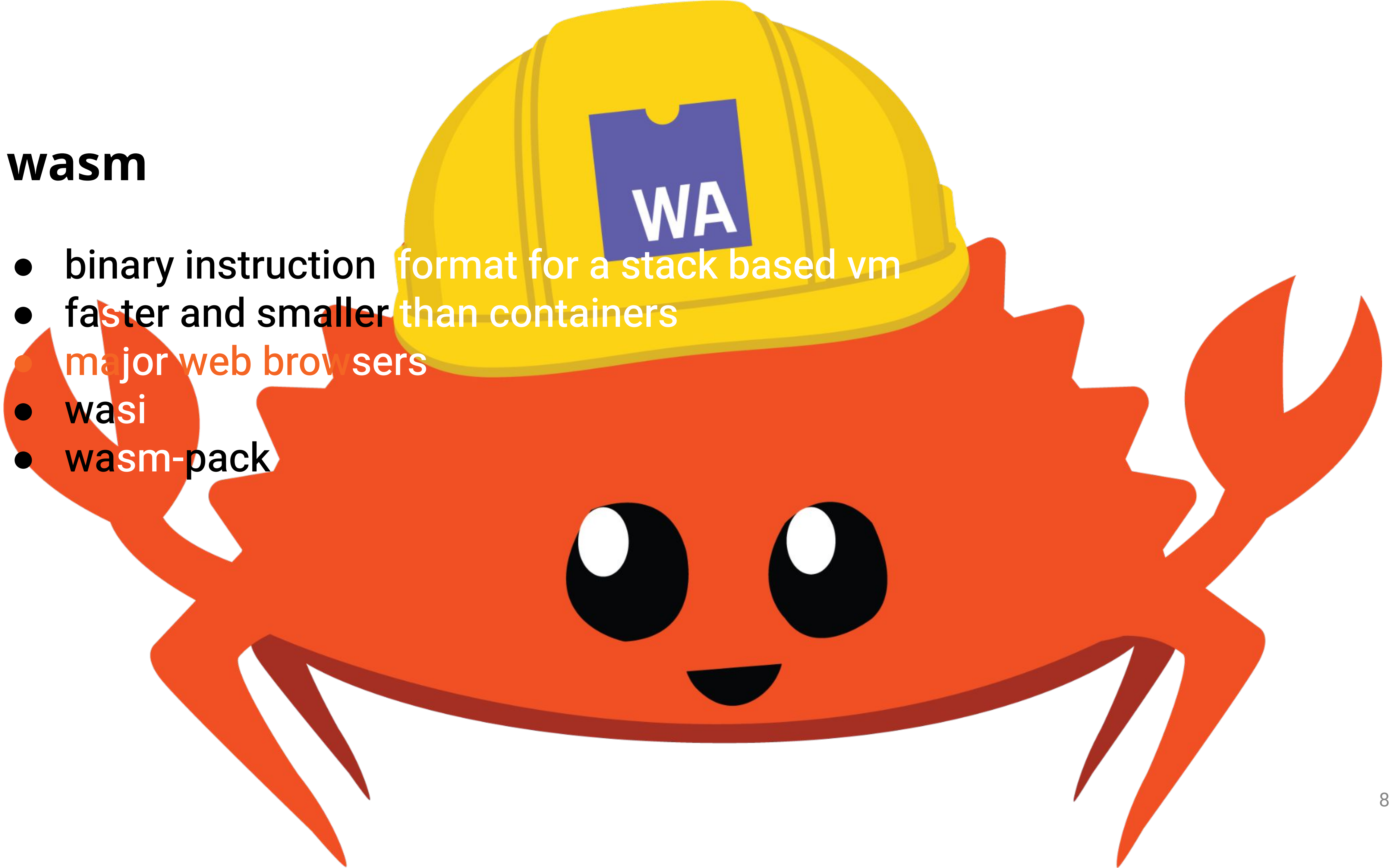

template
.Py
bundles
message queue
workers
time-series
running models

the bottleneck



source: https://developers.cloudflare.com/workers/learning/how-workers-works/

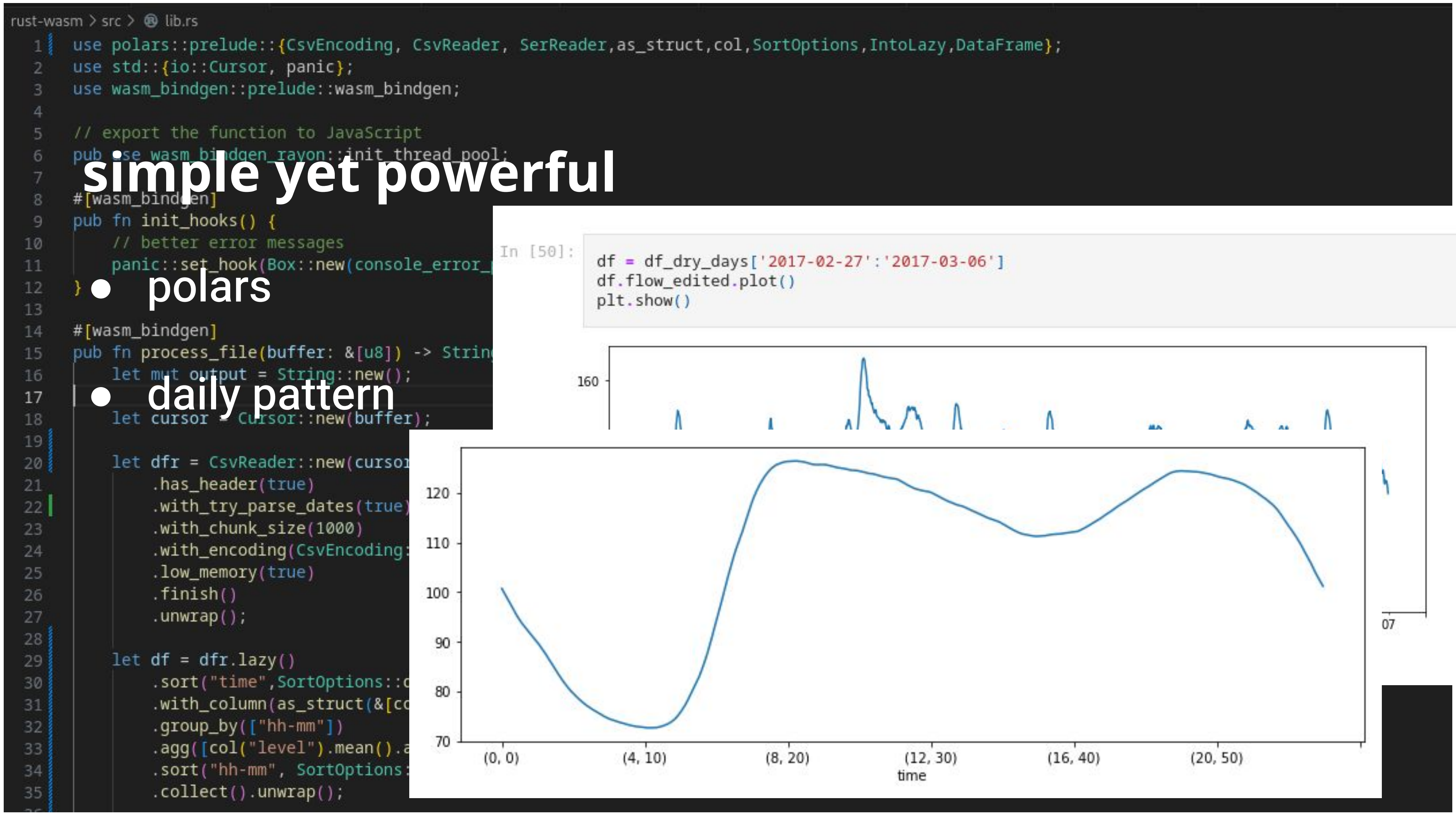
wasm



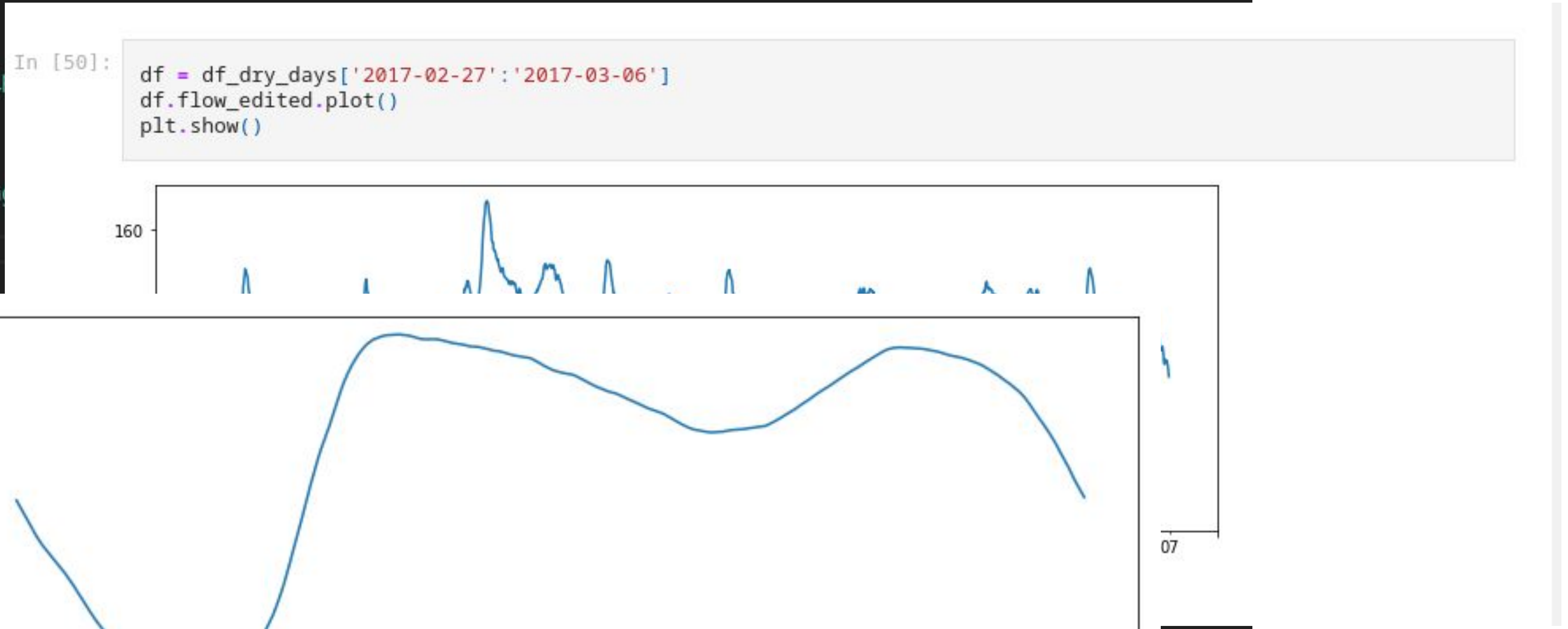
- binary instruction format for a stack based vm
- faster and smaller than containers
- major web browsers
- wasi
- wasm-pack

simple yet powerful

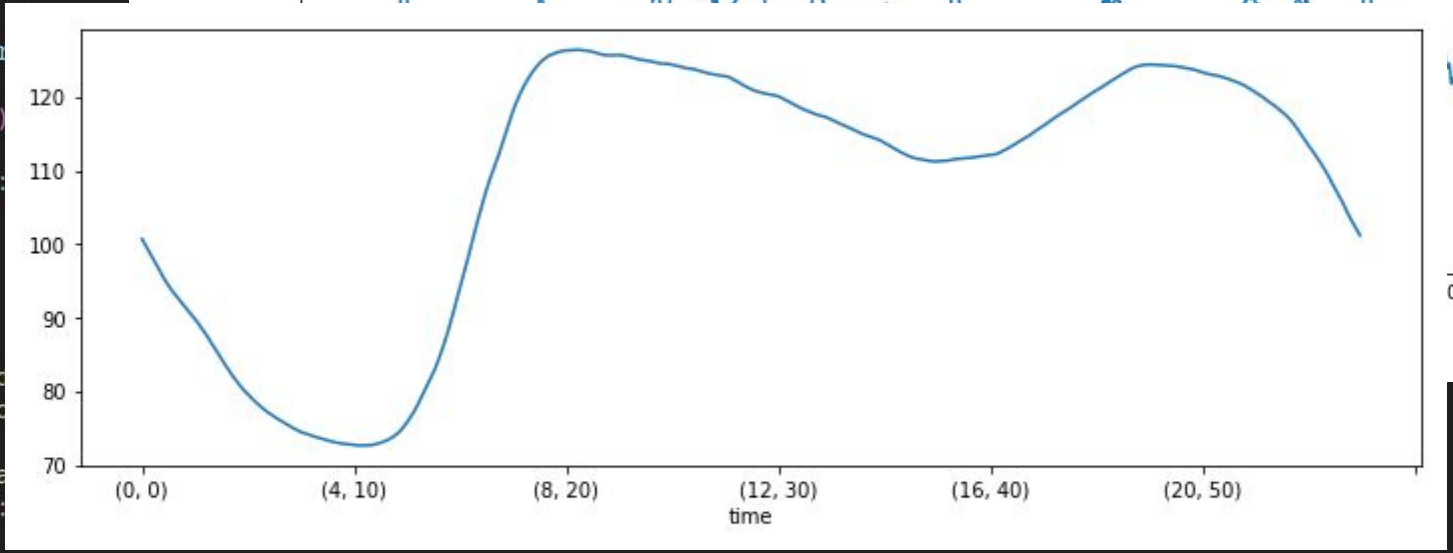
- polars
- daily pattern



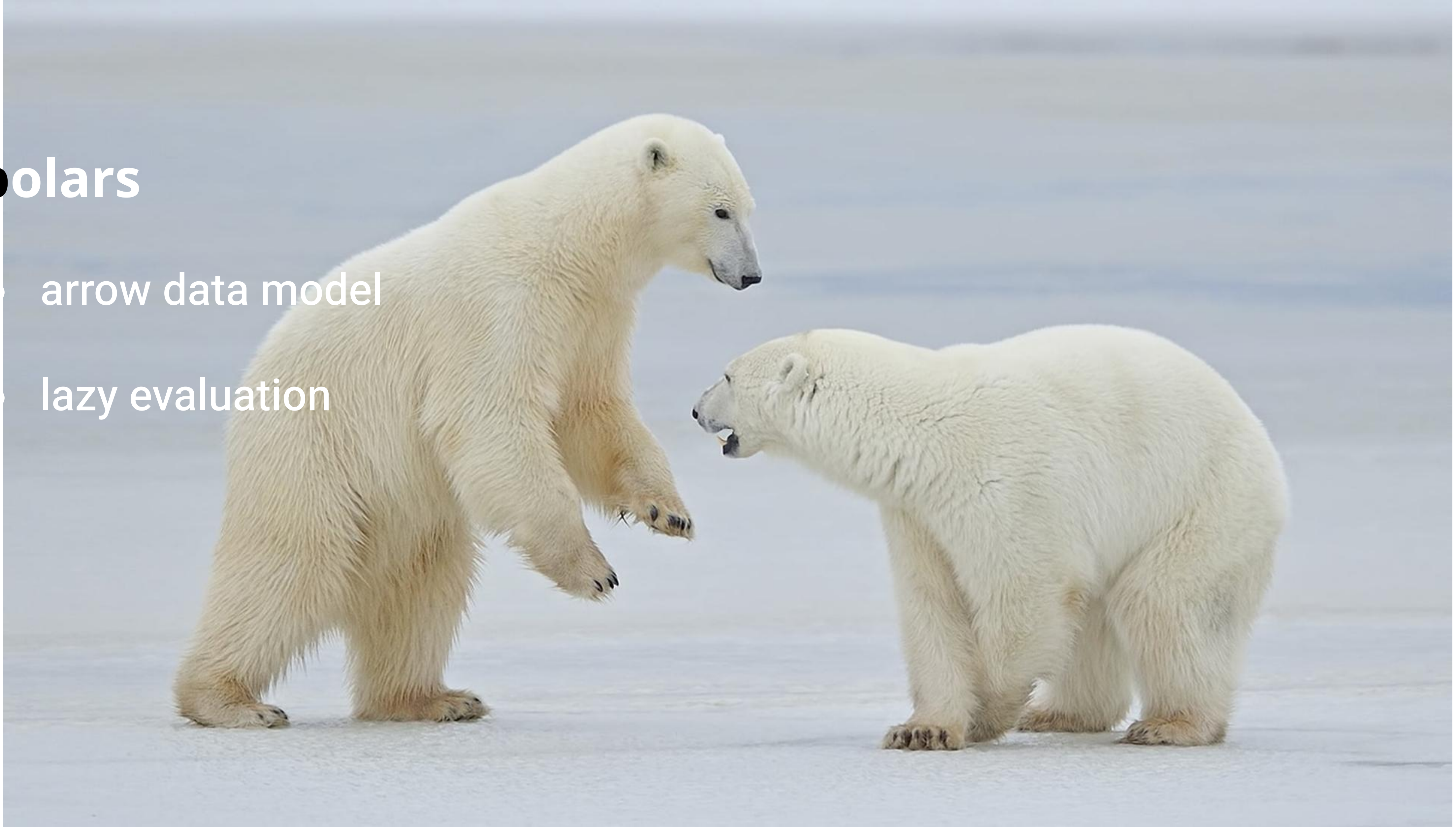
```
rust-wasm > src > lib.rs
use polars::prelude::{CsvEncoding, CsvReader, SerReader,as_struct,col,SortOptions,IntoLazy,DataFrame};
use std::{io::Cursor, panic};
use wasm_bindgen::prelude::wasm_bindgen;

// export the function to JavaScript
pub use wasm_bindgen_rayon::init_thread_pool;

#[wasm_bindgen]
pub fn init_hooks() {
    // better error messages
    panic::set_hook(Box::new(console_error_
}

#[wasm_bindgen]
pub fn process_file(buffer: &[u8]) -> Strin
    let mut output = String::new();

    let cursor = Cursor::new(buffer);

    let dfr = CsvReader::new(curso
        .has_header(true)
        .with_try_parse_dates(true
        .with_chunk_size(1000)
        .with_encoding(CsvEncoding:
        .low_memory(true)
        .finish()
        .unwrap();

    let df = dfr.lazy()
        .sort("time",SortOptions::c
        .with_column(as_struct(&[co
        .group_by(["hh-mm"])
        .agg([col("level").mean().a
        .sort("hh-mm", SortOptions:
        .collect().unwrap();
```



```
In [50]: df = df_dry_days['2017-02-27':'2017-03-06']
         df.flow_edited.plot()
         plt.show()
```



polars



- arrow data model
- lazy evaluation

torment of wasm

- python in wasm
- sockets
- embedding wasmtime in Rust
  - memory allocation
  - dependency hell

deus ex machina

Deno by Example

final remarks

- python in wasm
- still sockets
- parallel processing

@ github: stepinski

@ linkedin: stepinsky