

Playing the long game

A hobby project for 2029

Jon Skeet (he/him)

Context

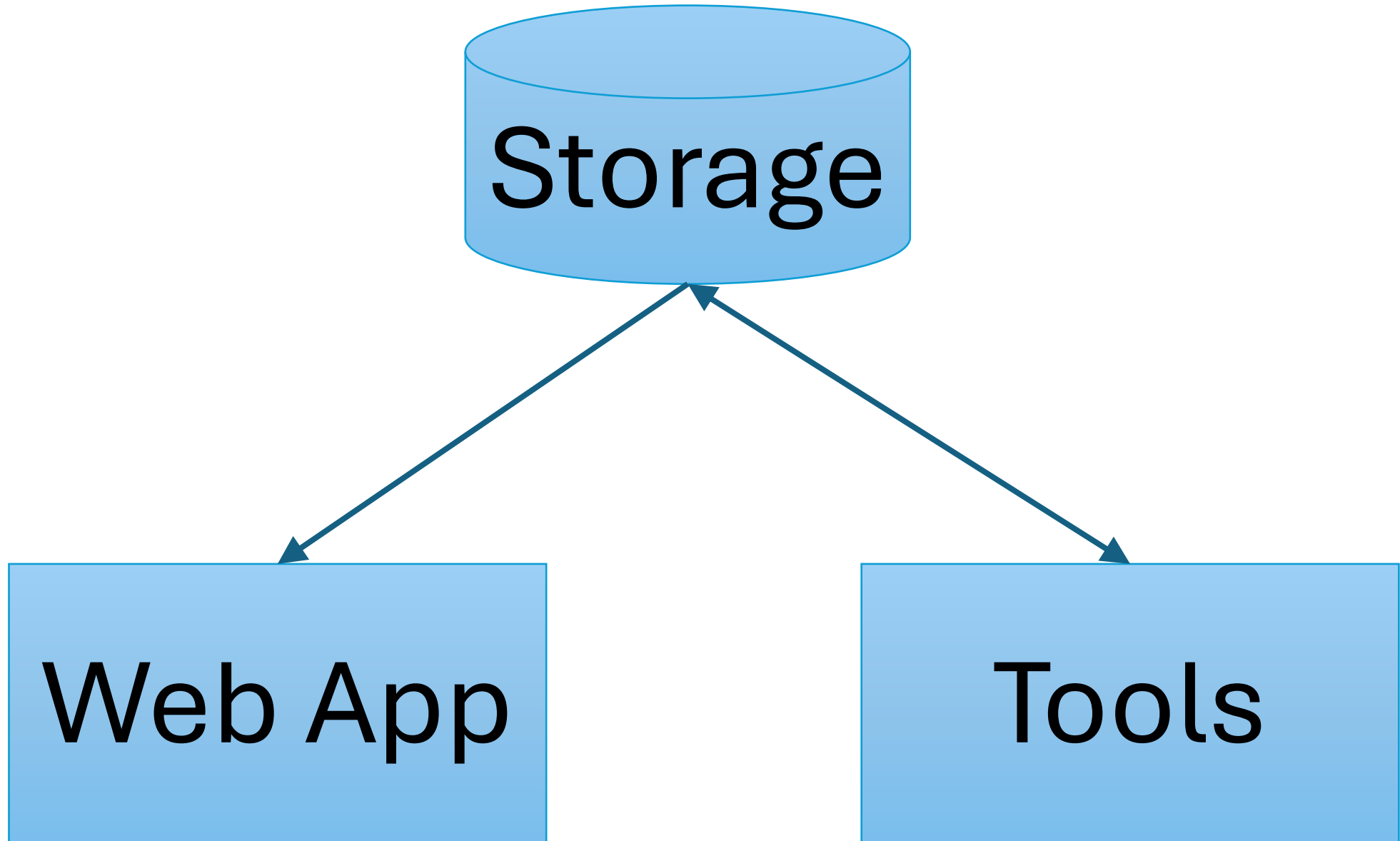
Goals (of the project)

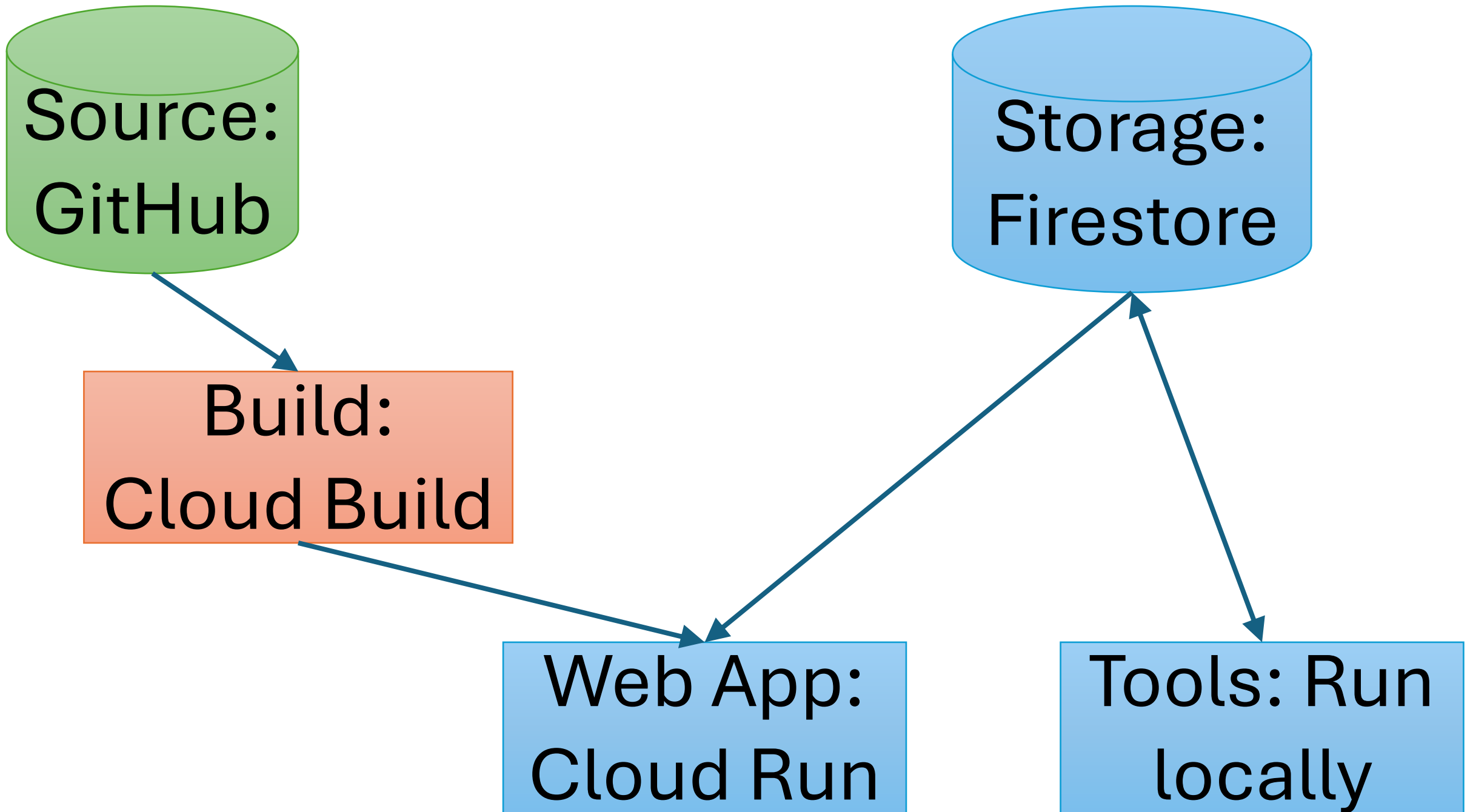
Goals (of this talk)

Site tour

(<https://election.uk>)

Architecture / Tech stack





Web technologies

- HTML (ASP.NET Core + Razor)
- CSS
- JavaScript (where necessary)
- JS libraries
 - plotly.js
 - oi.hexmap
- Some JSON APIs consumed by pages

Environments

Data design

What data do we need?

- Constituencies
- Parties
- Candidates
- Election results
- Polls
- Projection sets (per-constituency projections)
- Party changes
- Postcodes
- ... all in memory

Immutable types – but a lot of them

- Lots of records at a low level (Party, Constituency etc)
- Core vs non-core
 - Core types are the building blocks, independent
 - Non-core types can refer to core types (but not each other)
- “Raw” aggregation (nested record types):
 - ElectionCoreContextSourceData
 - ElectionContextSourceData
- More useful aggregation (classes) – used directly by tools and web:
 - ElectionCoreContext
 - ElectionContext
- More processed – used by web: AugmentedElectionContext

Home-grown MVVM

Data storage

Long time frames

Results and lessons

So much fun

Costs

Tools

The double-edged sword of records

- Records are simple immutable types
- Properties can be initialized “on construction”
- The `with` operator makes construction of new instances easy
- So what does this code do?

```
public sealed record Number(int Value)
{
    public bool Even { get; } = (Value & 1) == 0;
}
var n2 = new Number(2);
var n3 = n2 with { Value = 3 };
```

- See blog posts for more info (<https://codeblog.jonskeet.uk>)

Immutable types

- `ImmutableList => ImmutableArray`
- `ImmutableDictionary => FrozenDictionary`

Fun with postcodes

- ~2.7 million entries mapping postcode to constituency
- But with a 1MB storage restriction, and prefix search
- Example postcodes (outcode incode)
 - RG30 4TT
 - RG3 3AA
 - E1 1AA
 - E20 1AA
 - E1W 1AA
 - EC1A 1AA
- 4000 possible incodes per outcode
- Up to 8 constituencies per outcode

Fun with postcodes

- ~2.7 million entries mapping postcode to constituency
- But with a 1MB storage restriction, and prefix search
- Each outcode has a map; compress the lots

- Laptop benchmarks:

Search term	Time to search (ns)	Notes
RG	7836	17 constituencies
RG3	795	3 constituencies
RG30	443	3 constituencies
RG30 4	363	2 constituencies
RG30 4T	350	1 constituency
RG30 4TT	325	1 constituency
RG30 4AT	42	Doesn't exist

Next steps

Miscellaneous stats (1 week out of date...)

- ~27kloc C#
- ~2.5kloc Razor
- Sample page sizes (including everything)
 - Home: ~10k
 - Candidates: ~225k
 - Constituencies: 40k
 - Polls: ~100k (+492k of Plotly)
 - Seat map: ~10k (+7k of oi.hexmap)
- Load tests (causes scaling from 1->2 instances...)
 - 100+ qps of pages
 - 600+ qps of postcodes
 - No issues, and can probably take more