

The Case for Energy Clarity

The ability to understand application energy behavior in computer systems



Fan Chung



Henry Kuo



George Candea



3% of the world's energy consumption comes from data centers and computer networks, and the rise of ML will further drive this demand

But developers don't understand how the energy usage of their application is influenced by workloads, configurations, and other factors

To optimize energy costs, they need to know **why**, **where**, and **how** the energy is consumed!

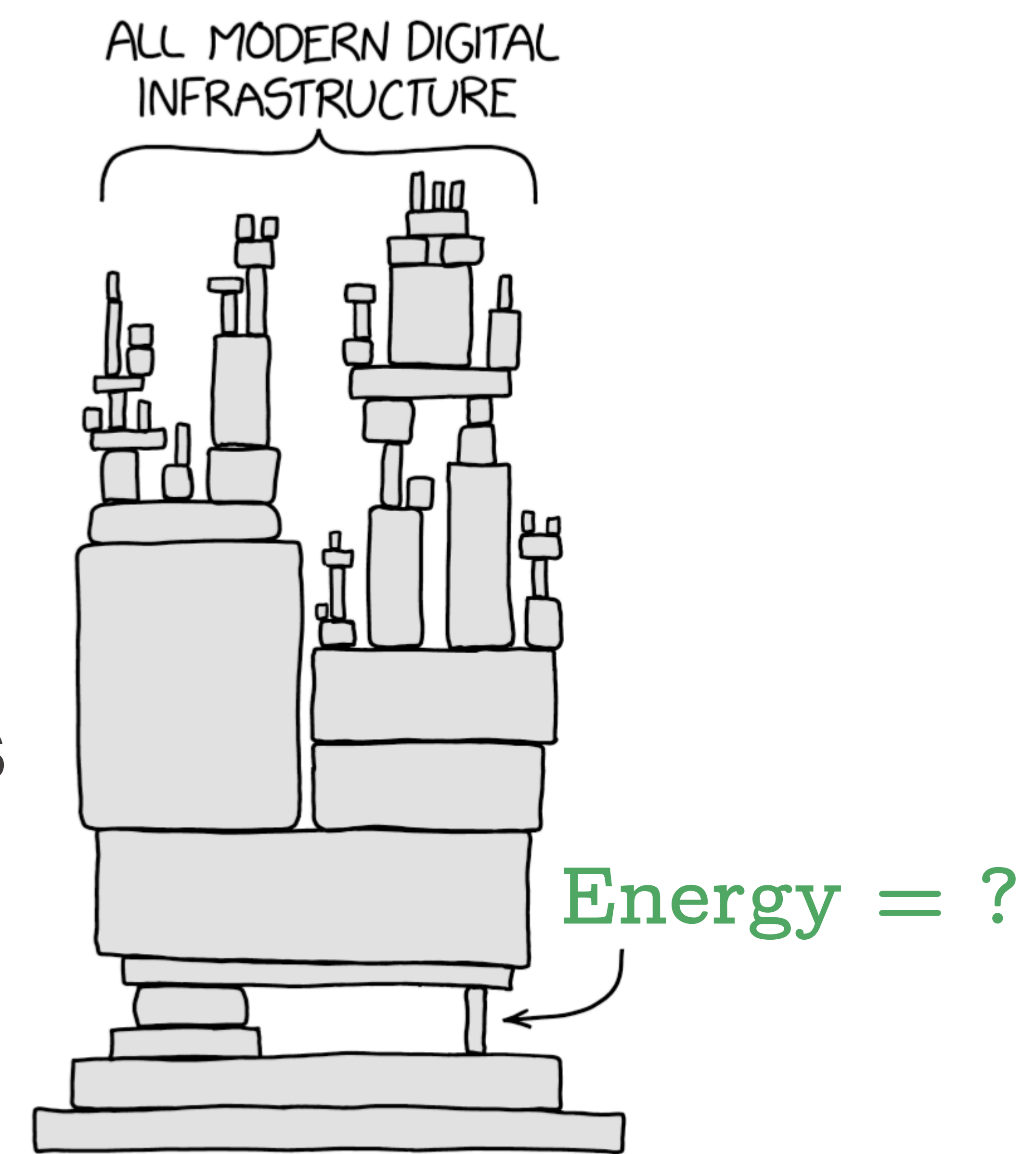


Figure: Modified from XKCD 2021: Software Development.

Solution: Energy Interfaces!

Energy interfaces are **programs** that describe application energy behavior, and they are

- **Actionable:** run to return energy
- **Readable:** match original app program structure
- **Complete:** estimate energy for all possible inputs

Energy interfaces can enable

- Energy-efficient ML inference
- Energy-aware task scheduling
-
- <Your own idea!>

```
@energy_interface
def energy_ml_webservice(request):
    if cache_exist(request):
        return energy_cache_fetch(request)
    else:
        return energy_cnn_forward(request.image)
```

```
@energy_interface
def energy_cache_fetch(request):
    return energy_redis_fetch(request.key)
```

```
@energy_interface
def energy_cnn_forward(image):
    return (
        8 * energy_conv2d(image.size())
        + 16 * energy_mlp(n_embedding)
        + 8 * energy_relu(n_embedding)
    )
```