



Shaping the hidden environmental impacts of software

GSHA 2023

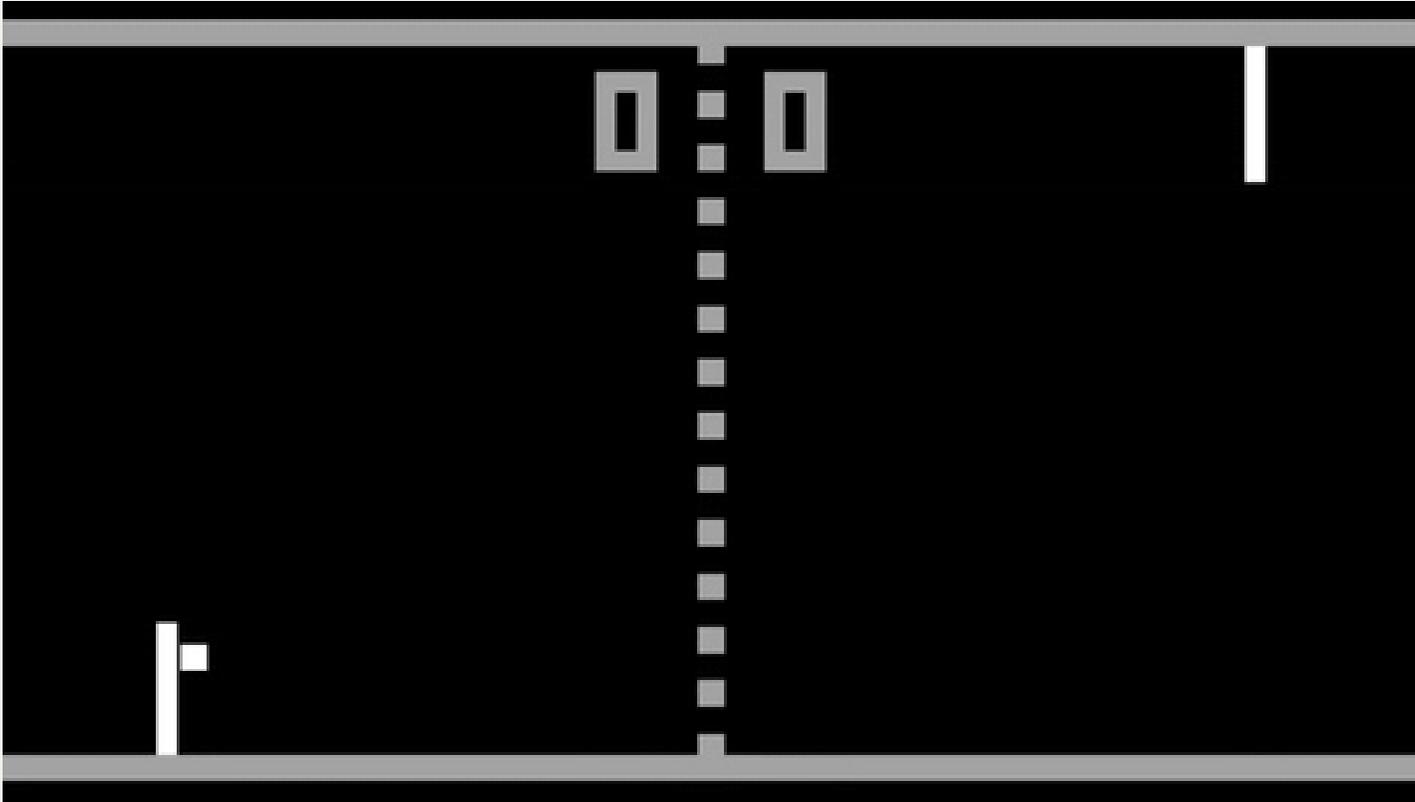
Thibault Simon, Pierre Rust, Romain Rouvoy

Software quickly became central



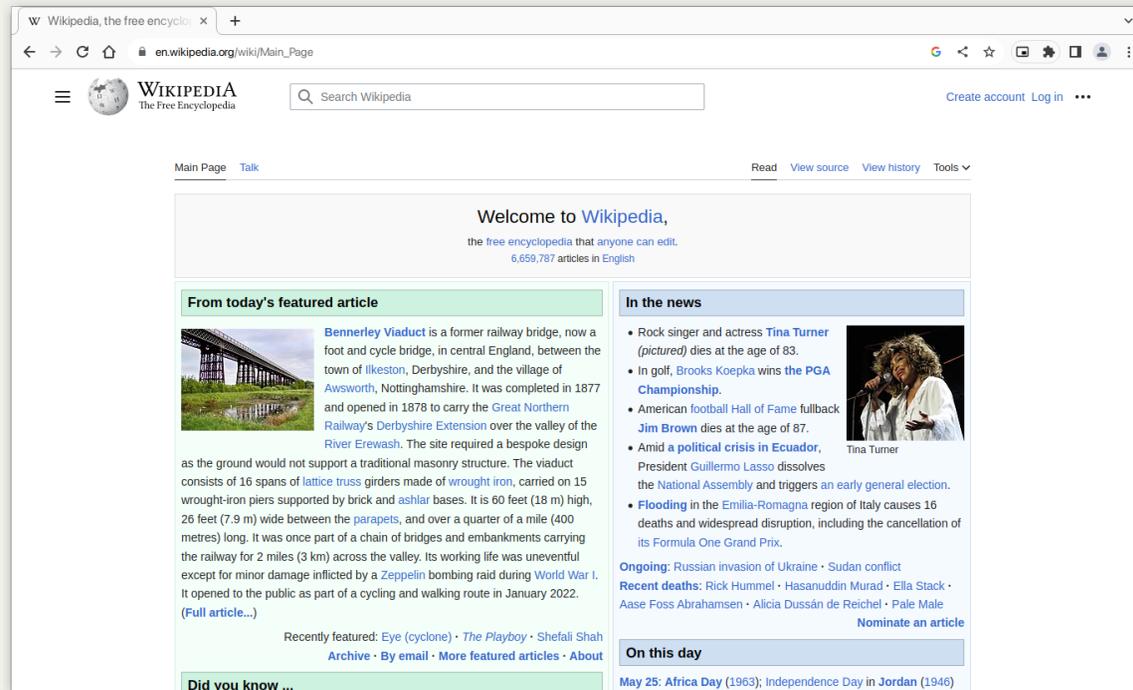
Why Software Is Eating The World (Marc Andreessen, WSJ, 2011)

Which led to increased complexity



Pong, 1972

Which led to increased complexity

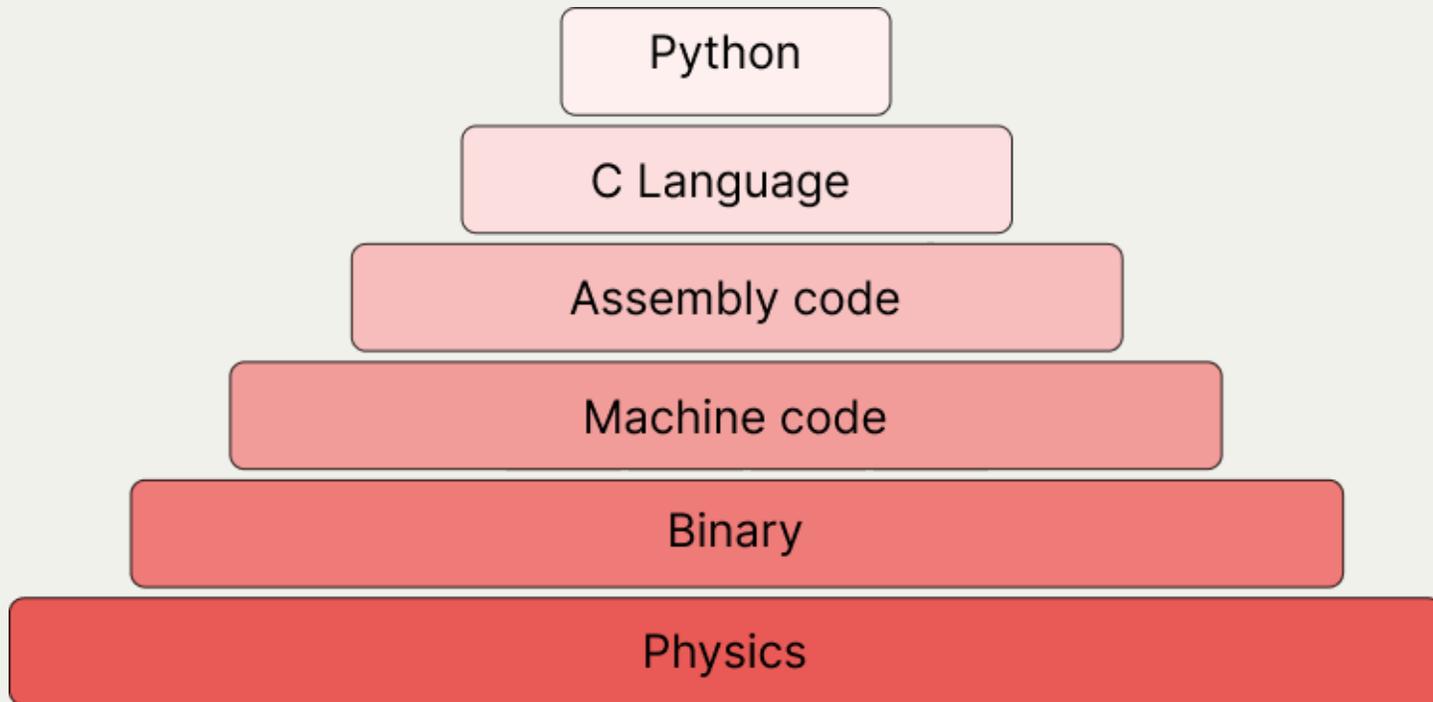


35 million lines of code¹

[1] <https://chromium.googlesource.com/chromium/src>

Hardware is complex

developers use different abstraction levels



Software scale grew *a lot*

2.99 billion monthly active users¹

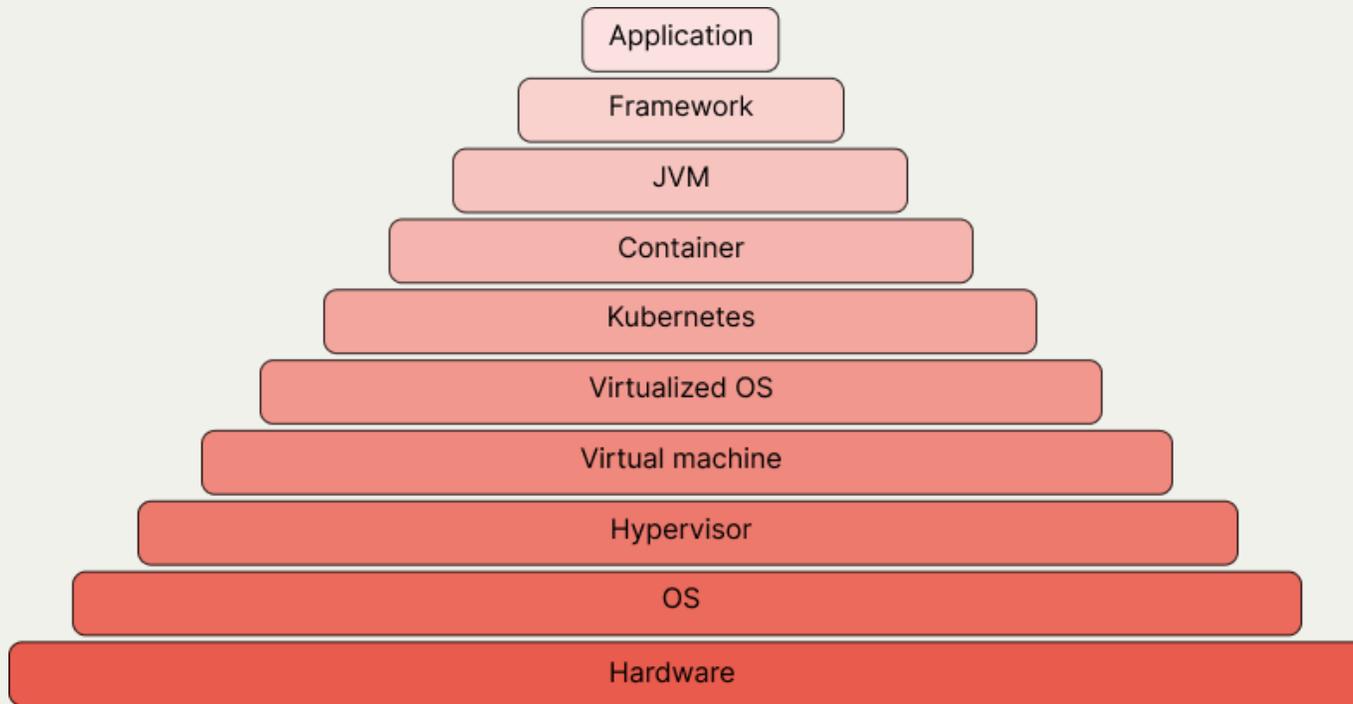


Visualizing Friendships, Facebook engineering, 2010

[1] First Quarter 2023 Financial Highlights, Meta

Software scale grew *a lot*

Even more abstraction layers were added



Which **distribute** responsibility (*separation of concerns*)

Which **distribute** responsibility (*separation of concerns*)
and **dilute** the underlying infrastructure's environmental
footprint

While ICT's environmental footprint comes from hardware,
responsibility lies within software

Energy consumption is driven by software

Between 4 and 8% of global electricity use^{1, 2}



[1] J. Malmodin and D. Lundén, *"The Energy and Carbon Footprint of the Global ICT and E&M Sectors 2010-2015,"* Sustainability, vol. 10, no. 9, p. 3027, Aug. 2018

[2] A. SG Andrae, *"New perspectives on internet electricity use in 2030,"* Engineering and Applied Science Letter, vol. 3, no. 2, pp. 19-31, 2020.

Device manufacturing is driven by software



Mineral extraction has contributed to environmental degradation, population displacement, violent conflicts and human rights violations¹

[1] *Mapping Mining to the SDGs: An Atlas*, UN, 2016

Software can drive e-waste generation

Up to 20% of device renewal in EU are caused by software¹



Credit: Muntaka Chasant

[1] Obsolescence logicielle, Rapport CGEDD, 2020

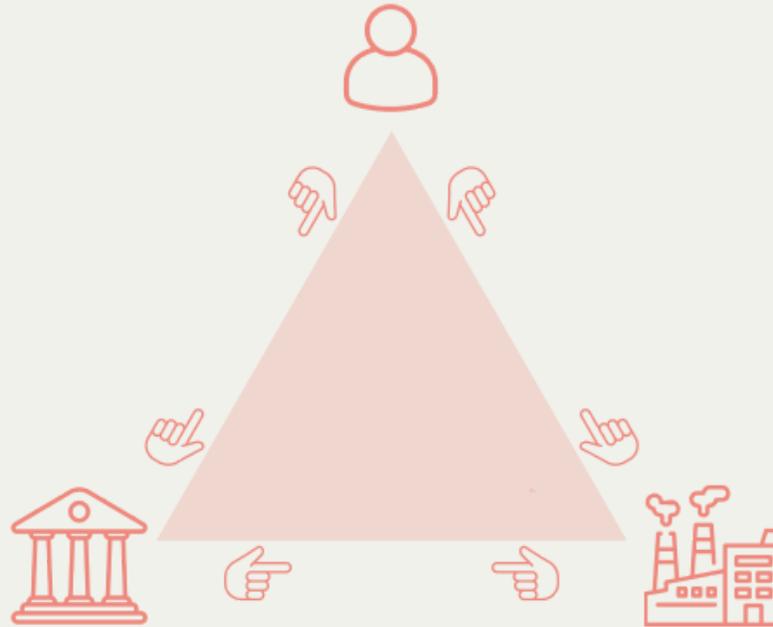
Can I, as a software engineer, have a direct effect on the repairability of hardware devices? On the tracing of e-waste?
On the social consequences of mining?

Can I, as a software engineer, have a direct effect on the repairability of hardware devices? On the tracing of e-waste? On the social consequences of mining?





Assigning responsibility to others does not trigger actions



Adapted from *The Triangle of Inaction*, Pierre Peyretou

What is my responsibility as a software engineer?

What is my responsibility as a software engineer?
to act *at my level*

The environmental impact levels of ICT^{1,2}:

1. **Direct** effects: negative environmental impacts of ICT hardware life cycle
2. **Indirect** or enabling effects: optimization, obsolescence...
3. **Systemic** effects: long-term reaction (notably rebound effects)

[1] Berkhout, F., & Hertin, J. (2001). *Impacts of information and communication technologies on environmental sustainability: Speculations and evidence*. Report to the OECD, Brighton, 2001.

[2] L. M. Hilty and B. Aebischer, *ICT for Sustainability: An Emerging Research Field*, in *ICT Innovations for Sustainability*, 2015, pp. 3-36.

Indirect effects depend on external factors out of developer's
control¹

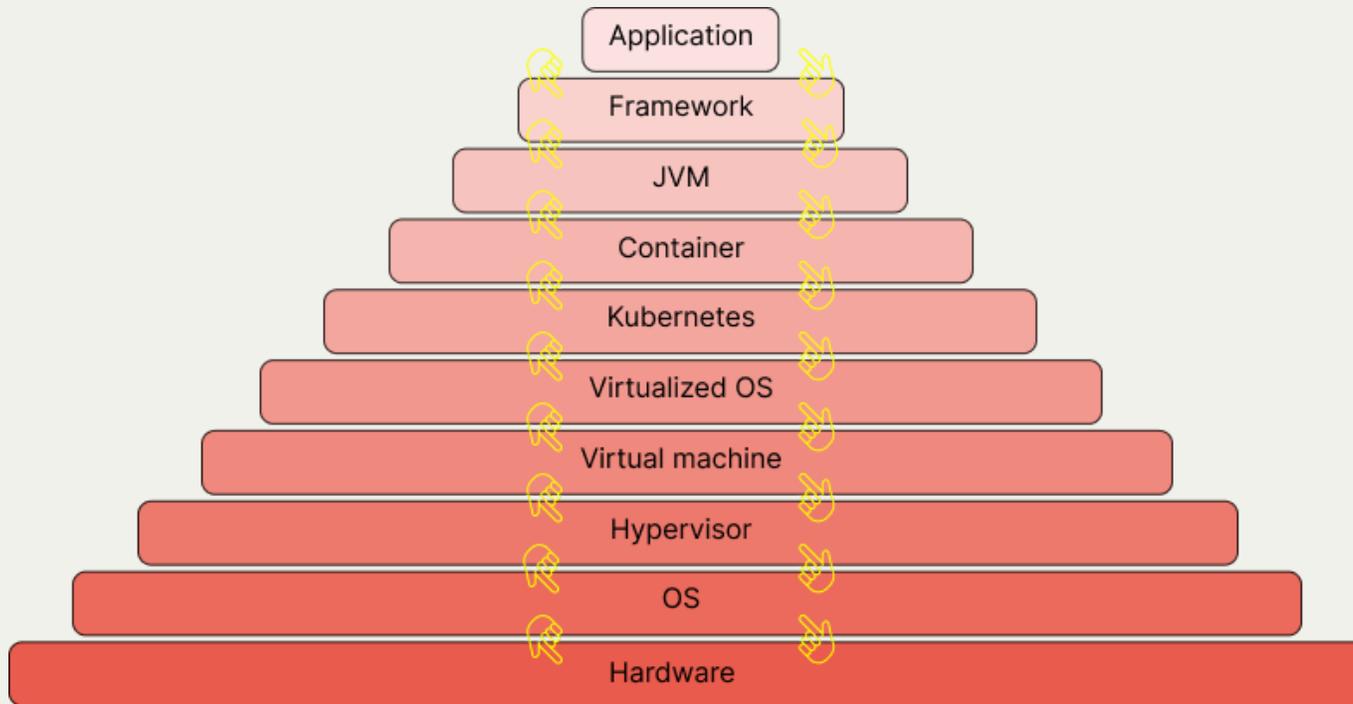
[1] Calero, Coral, et al. *Quality in Use and Software Greenability*. RE4SuSy@RE. 2014.

Developers should focus on lowering their software's **direct**
impact

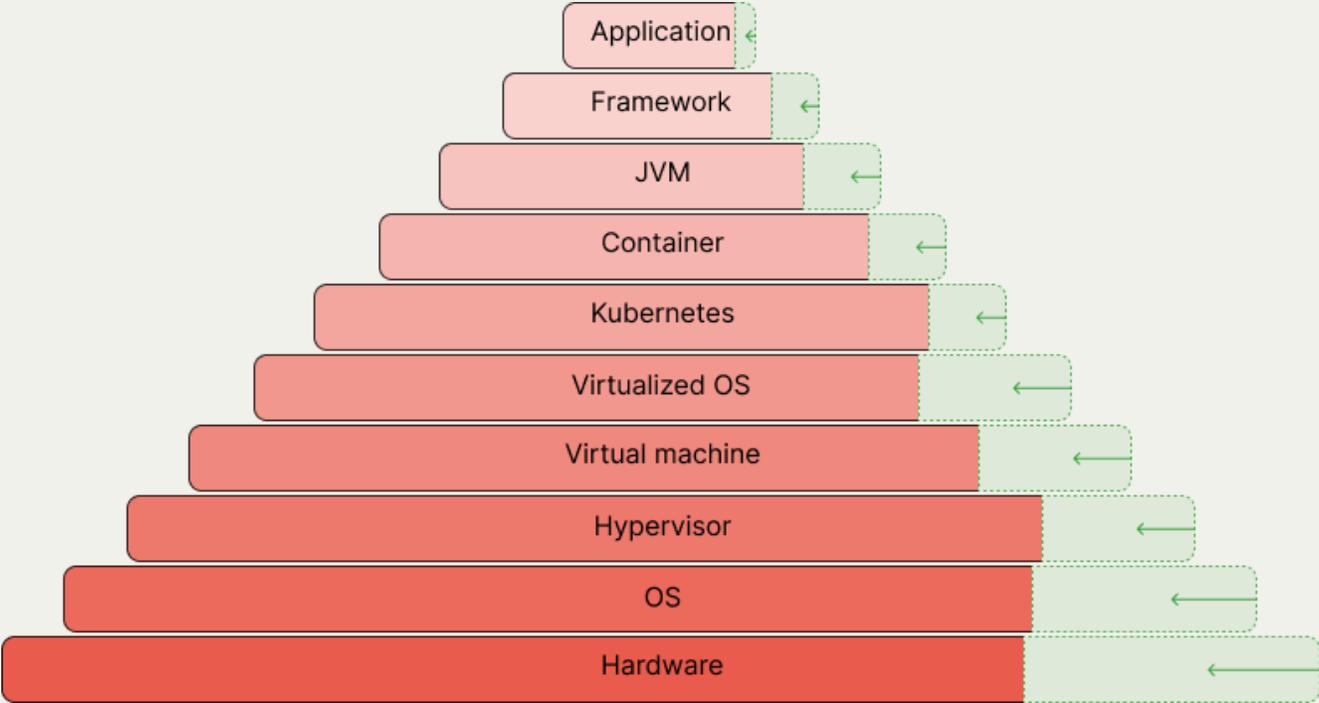
Developers should focus on lowering their software's **direct**
impact

by lowering the pressure they place on **underlying layers**

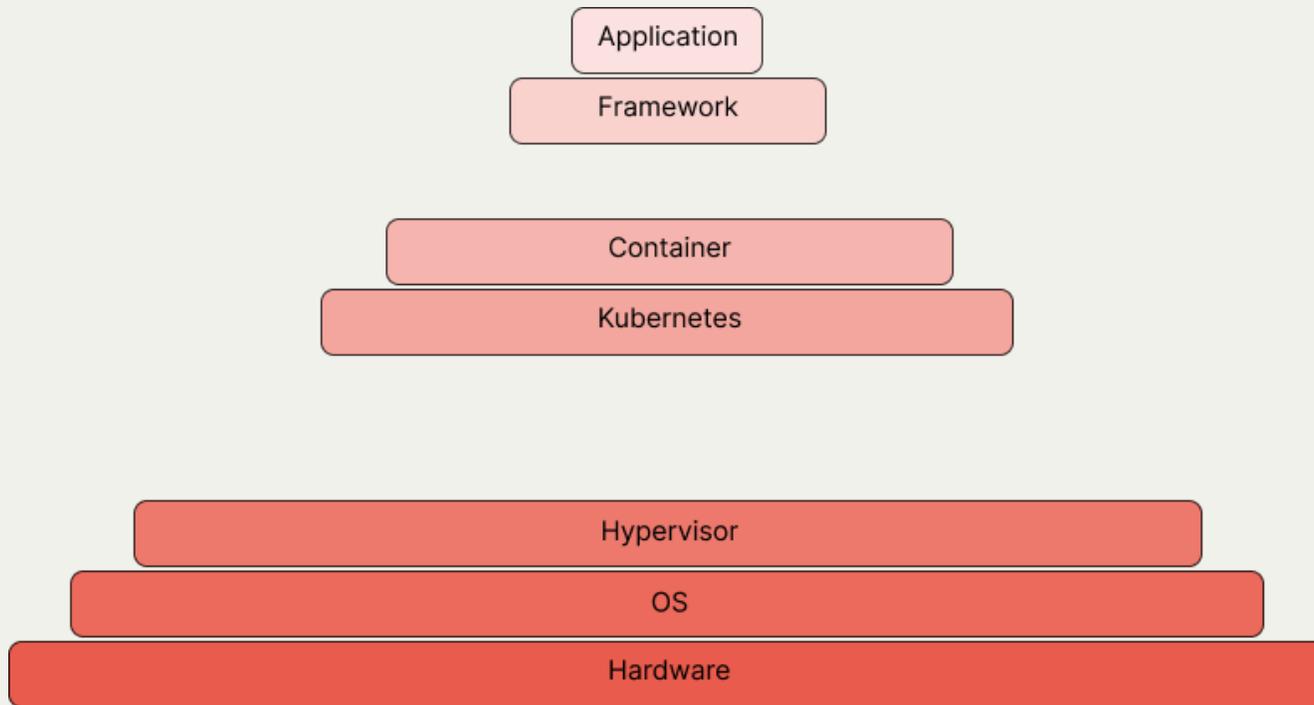
Avoid falling into an *inaction loop*



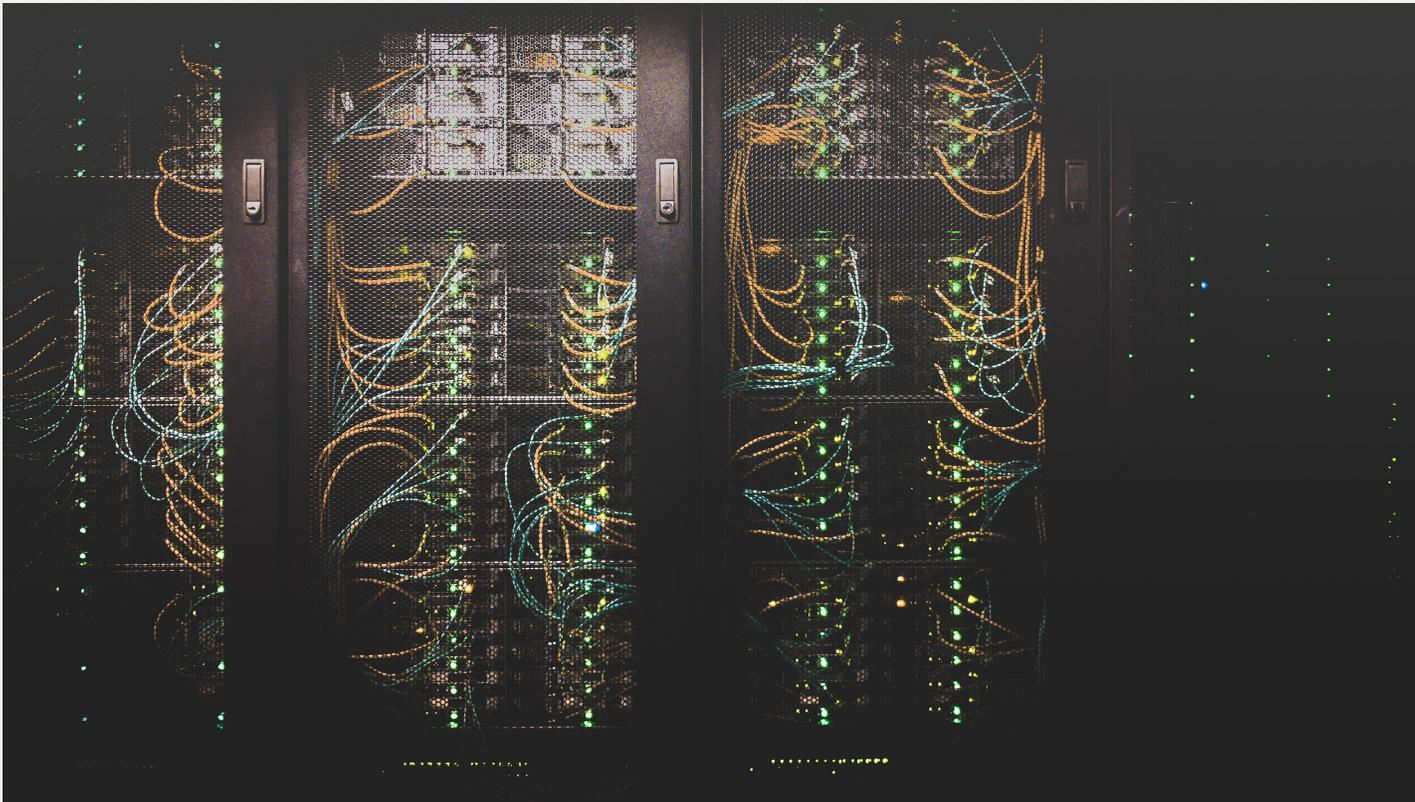
Monitor and reduce resources *used* and *reserved*



Challenge the need for abstraction layers



Virtual resources from any abstraction layer can be linked to hardware



Which can be translated to an environmental impact over multiple categories through:

- Energy consumed
- Hardware embodied emissions



Which can be translated to an environmental impact over multiple categories through:

- Energy consumed
- Hardware embodied emissions



To measure, understand and lower it

Conclusion

While software is virtual, its environmental footprint is not

Conclusion

While software is virtual, its environmental footprint is not

If you wait for underlying components to reduce their footprint, they will do the same

Conclusion

While software is virtual, its environmental footprint is not

If you wait for underlying components to reduce their footprint, they will do the same

We need to go further than energy consumption and co2e emissions

Future work

Holistic approach on software life cycle¹

[1] *Uncovering the Environmental Impact of Software Life Cycle*. Preprint: <https://hal.science/hal-04082263/>

Future work

Holistic approach on software life cycle¹

Lowering resources waste at all layers

[1] *Uncovering the Environmental Impact of Software Life Cycle*. Preprint: <https://hal.science/hal-04082263/>

Thank you for your attention
Any questions ?

