

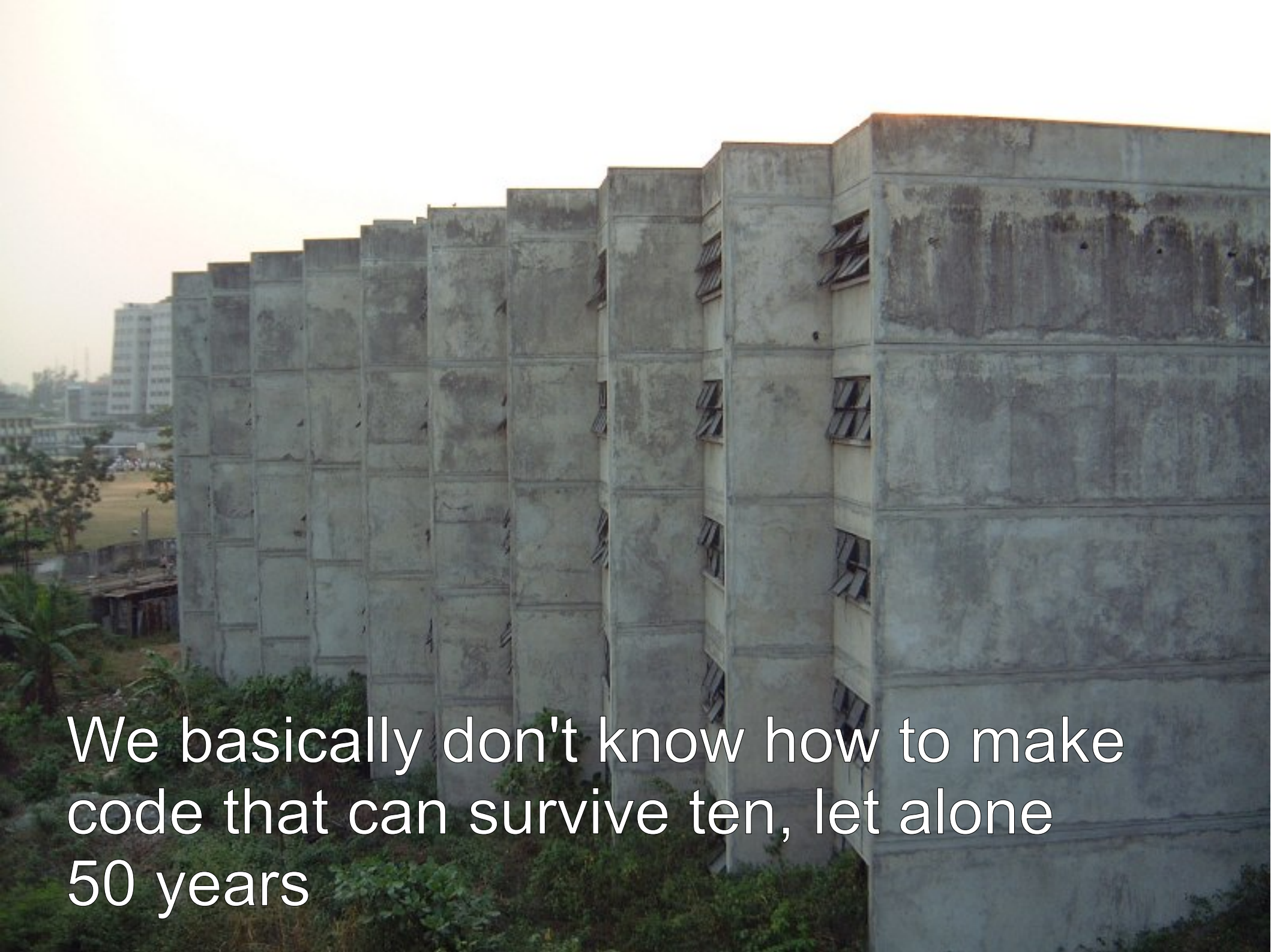
Software Architecture using ØMQ



Pieter Hintjens
#techmesh 2013



90% of software is already trash,
90% of the rest will be, soon



We basically don't know how to make
code that can survive ten, let alone
50 years



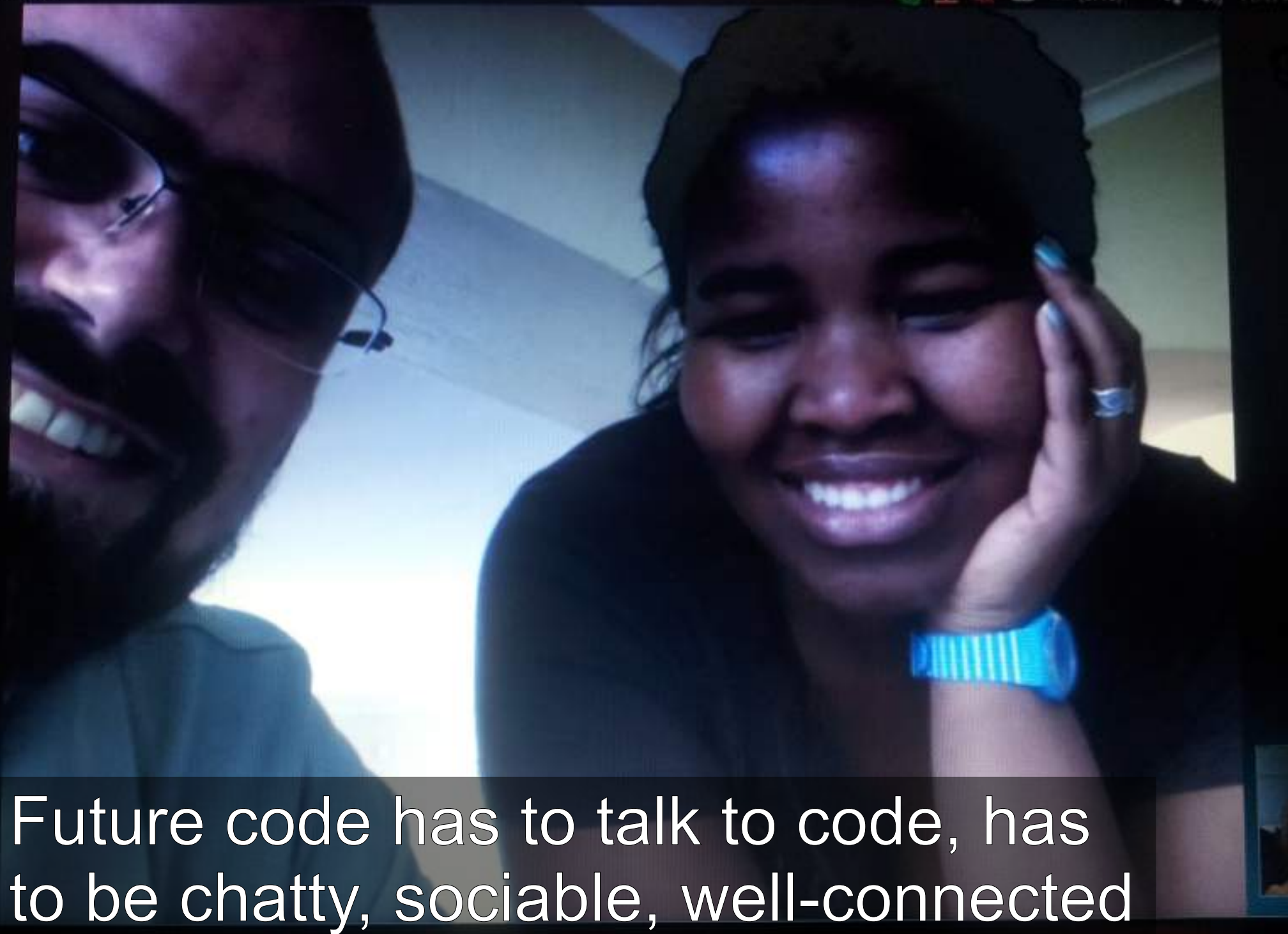
Software architecture should be answering this question...



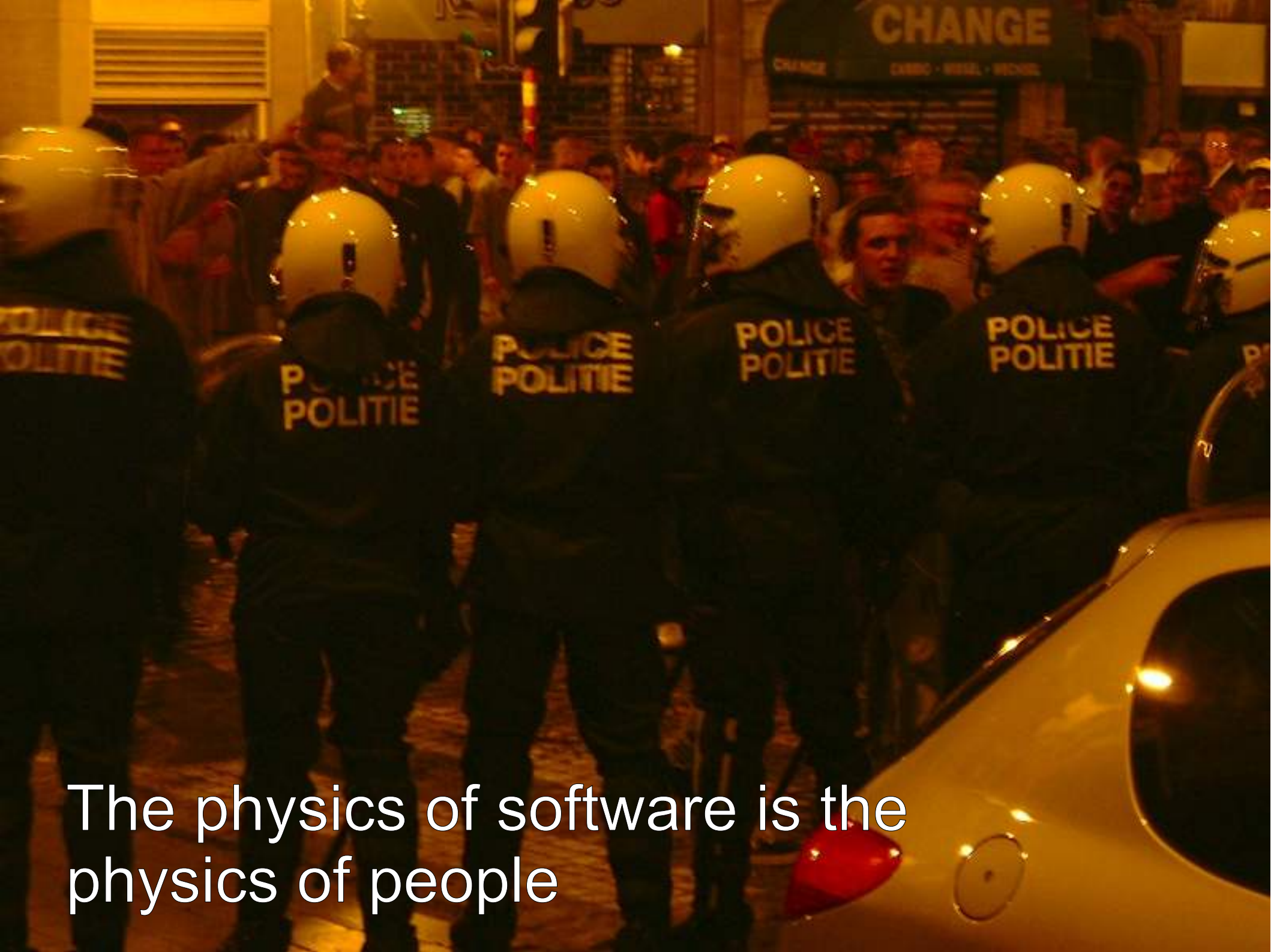
Can we build durable, perfect large-scale systems?



The answer is all around us...



Future code has to talk to code, has to be chatty, sociable, well-connected



The physics of software is the
physics of people



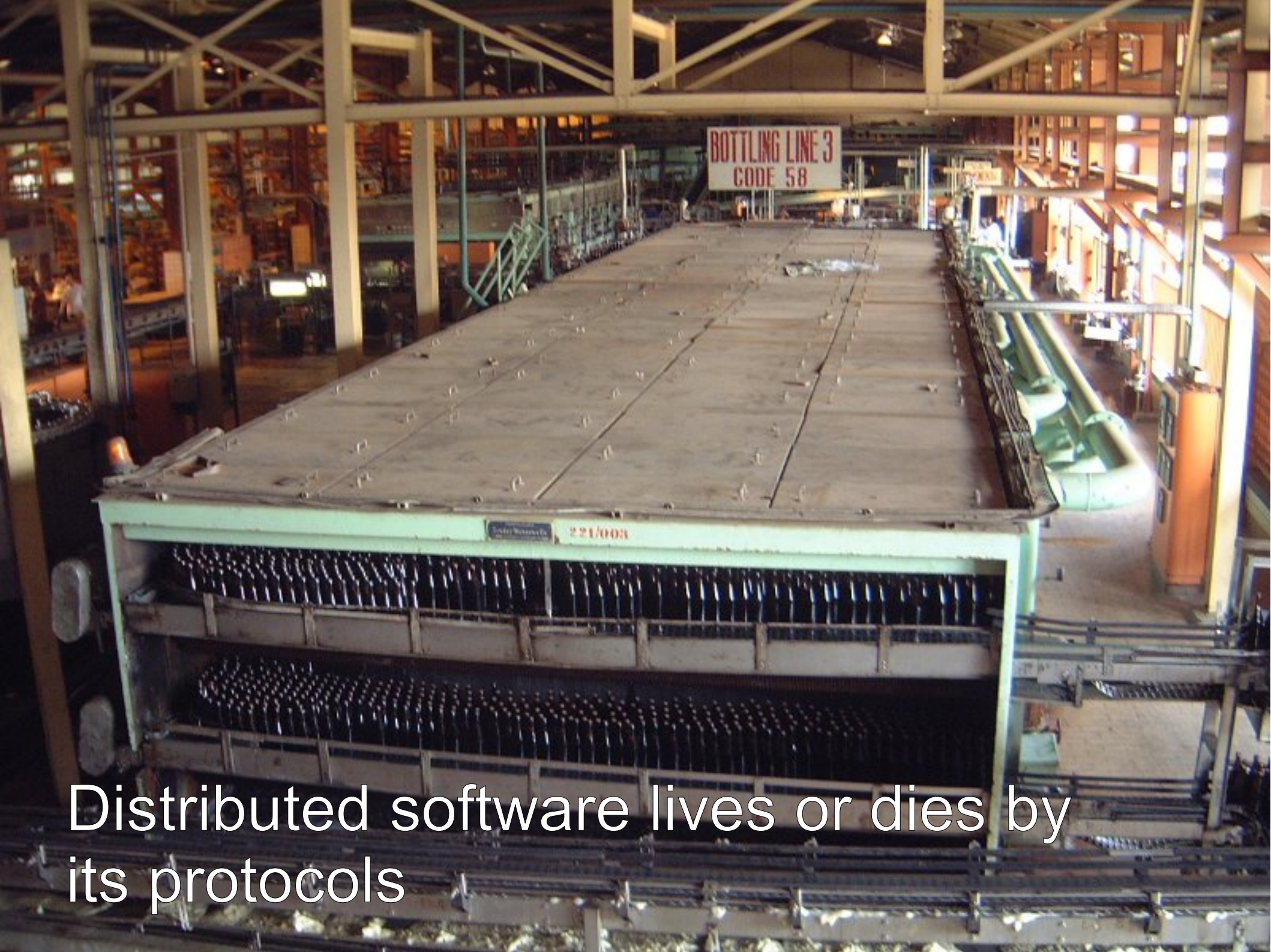
How we connect to each other
matters more than who we are

A photograph of a live jam session on a stage. The scene is bathed in warm, orange and yellow light, creating a hazy, energetic atmosphere. In the foreground, a person is seated at a drum kit, their hands positioned over the drums. To their right, another person stands, playing a guitar. In the background, other musicians are visible, including one who appears to be singing into a microphone. The overall composition is dynamic and captures the essence of a spontaneous musical performance.

Writing distributed code is like a live
jam session, all about other people




A good contract is worth a thousand assumptions



Distributed software lives or dies by
its protocols



Protocols define the rights and obligations of each party

The image displays four circular targets arranged in a 2x2 grid. Each target has a blue center with concentric white rings. The top-left target shows high precision and high accuracy, with all shots clustered tightly in the center. The top-right target shows high precision but low accuracy, with all shots clustered together but in the lower-right quadrant. The bottom-left target shows low precision and low accuracy, with shots scattered widely across the target. The bottom-right target shows low precision and high accuracy, with shots scattered widely but centered around the bullseye. A semi-transparent grey box with white text is overlaid on the bottom-left target.

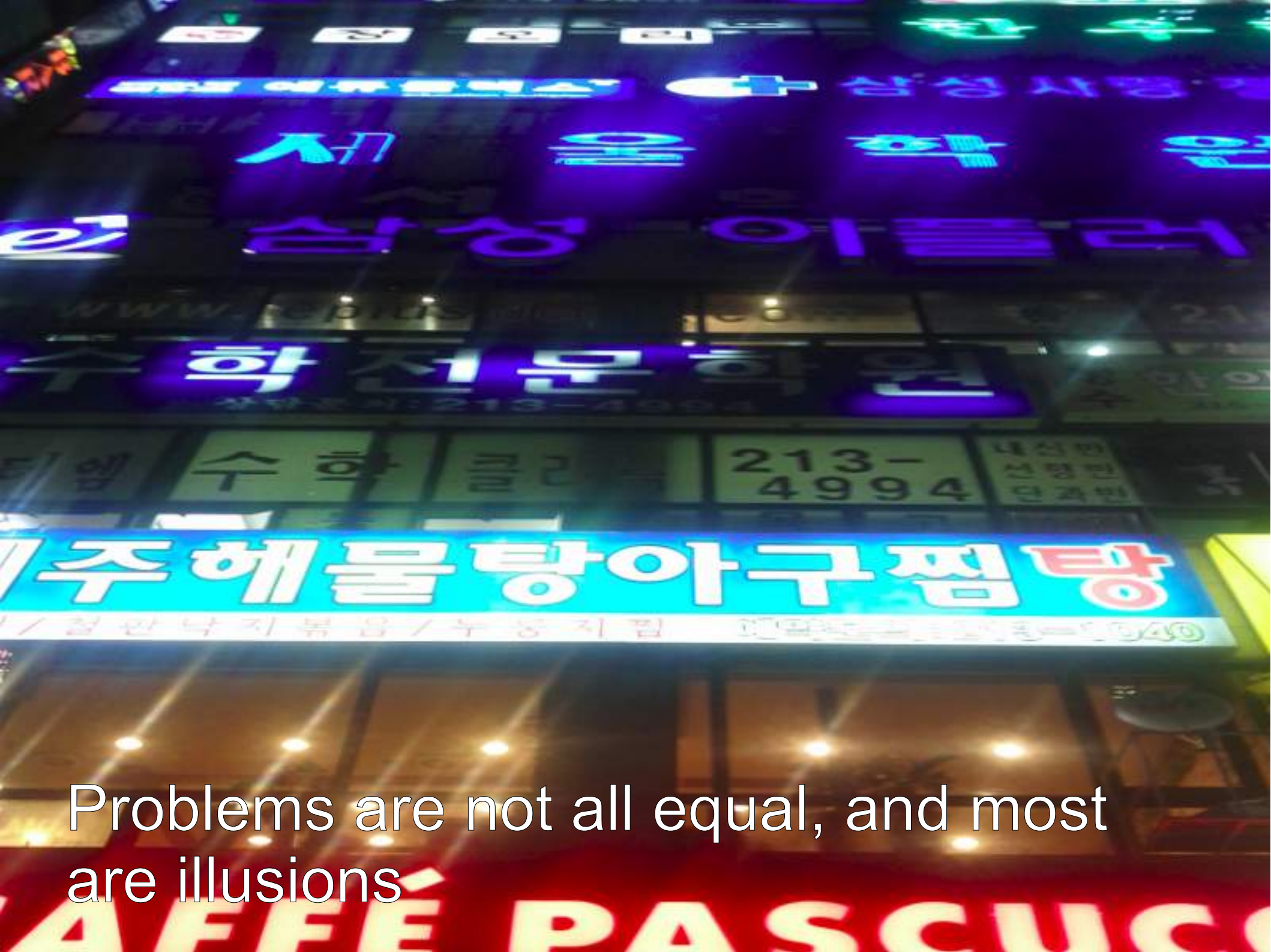
Extensibility and performance are
two separate targets



Simplicity always beats functionality

A man in a light blue shirt and shorts stands on a grassy hill, looking out at a vast landscape under a blue sky with large white clouds.

Design by removing problems, not
adding features



Problems are not all equal, and most
are illusions

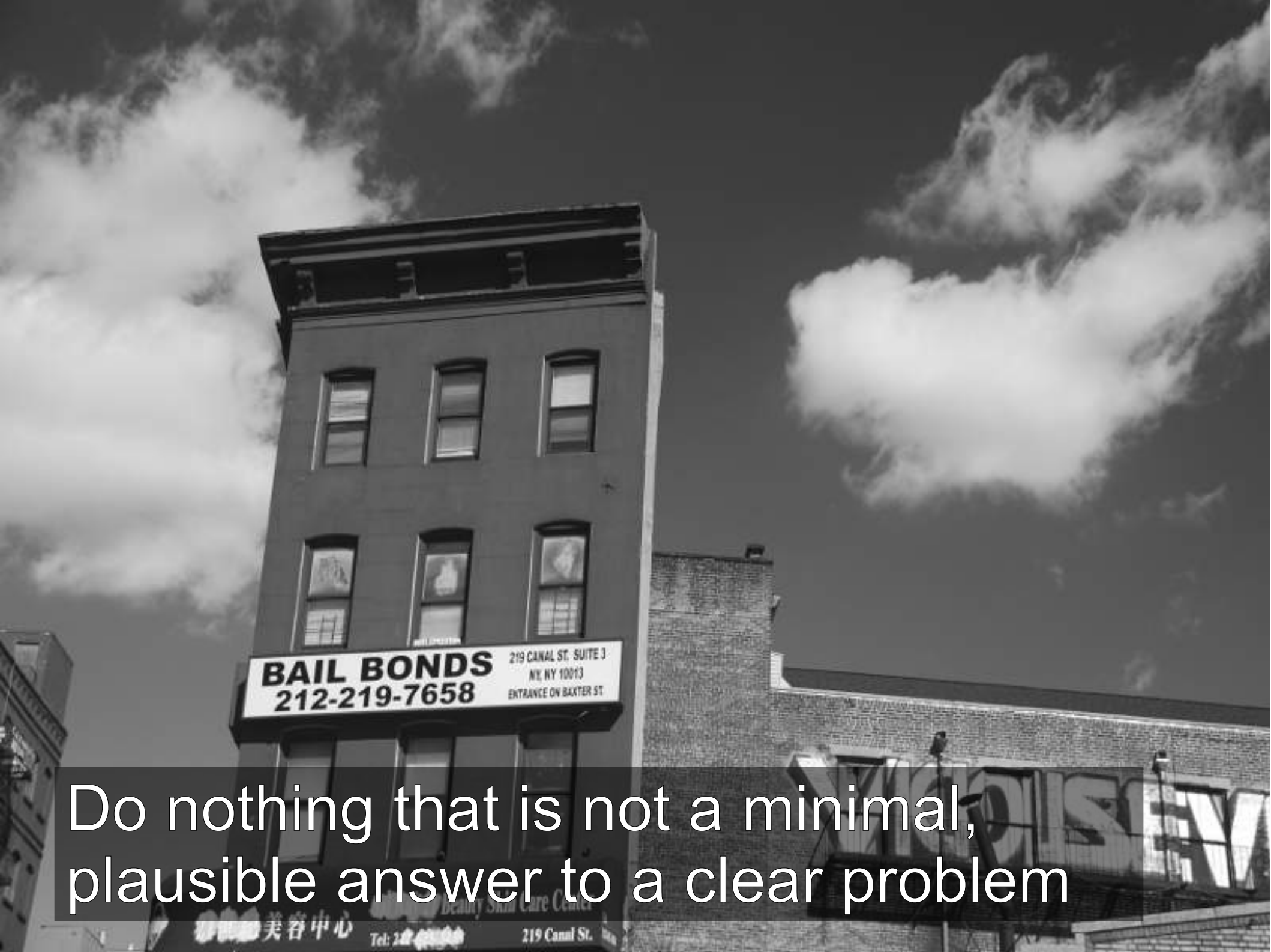
CAFFÉ PASCUCCI



When you know the real problem you
have done half the work



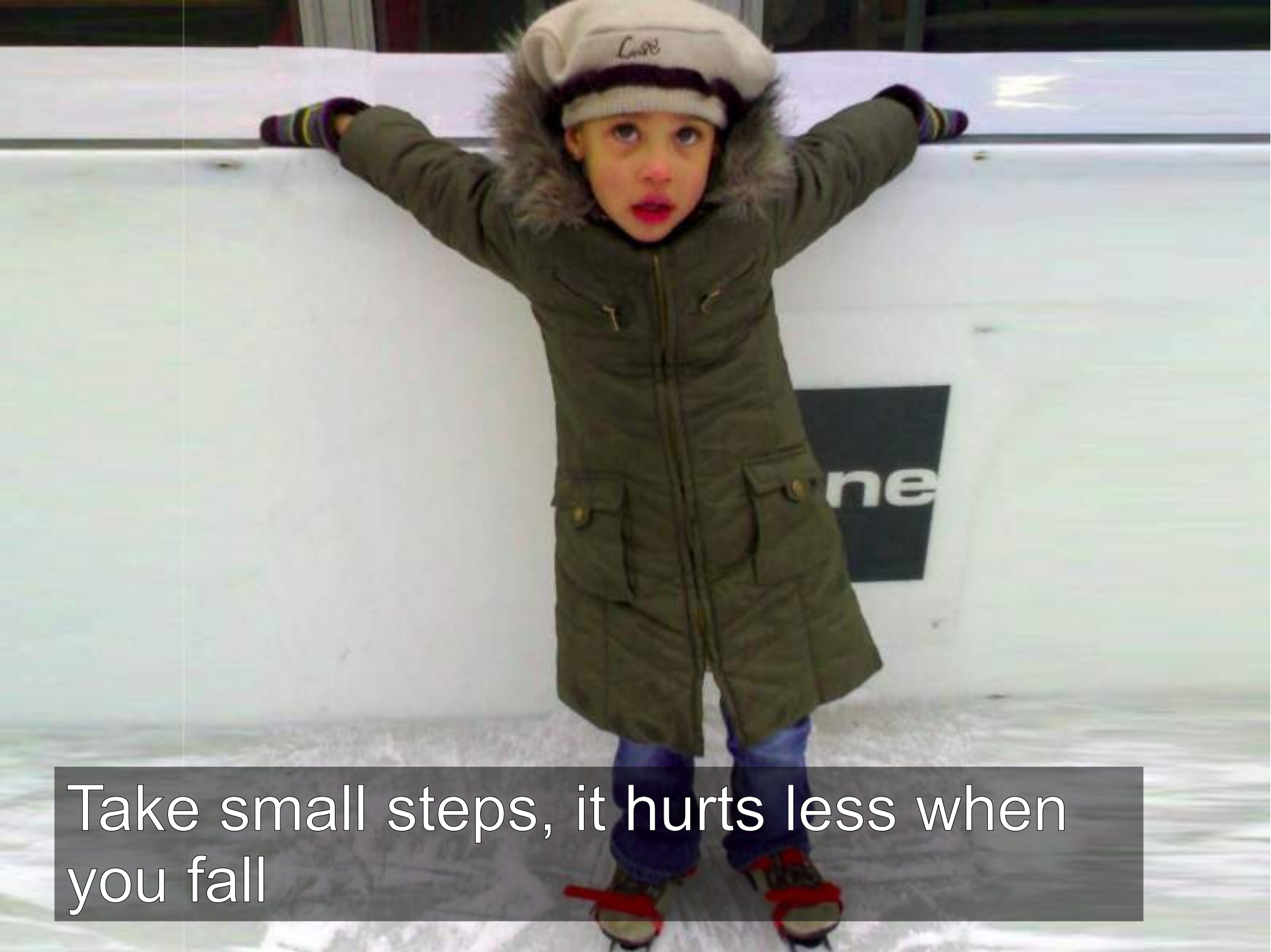
The problem landscape is real, and discoverable over time



Do nothing that is not a minimal,
plausible answer to a clear problem



Make every commit shippable



Take small steps, it hurts less when you fall



Solve one problem, and repeat until
you run out of time or money

Learn more:

The ØMQ Guide <http://zero.mq/zg>
Coming from **O'Reilly** in Feb 2013

ØMQ London <http://zero.mq/lug>
SkillsMatter eXchange, today 6.30pm