power consumption and how many you could cram in a rack. In this world we built banks of compute and created virtual machines as we needed them. Then we got public utility forms with the arrival of AWS EC2 in 2006.

The more industrialised forms of any activity have different characteristics to early evolving versions. With computing infrastructure then utility forms had similar processing, memory and storage capabilities but they had very low MTTR. When a virtual server went bang, we didn't bother to try and fix it, we didn't order another, we just called an API and within minutes or seconds we had a new one. Long gone were the days that we lovingly named our servers, these were cattle not pets.

This change of characteristics enabled the emergence of a new set of architectural principles based upon a low MTTR. We no longer cared about N+1 and resilience of single machines, as we could recreate them quickly if failure was discovered. We instead designed for failure. We solved scaling by distributing the workload, calling up more machines as we needed them — we had moved from scale up to scale out. We even reserved that knowing chortle for those who did "capacity planning" in this world of abundance.

Figure 98— Emergence of a new practice