note that you don't need to examine the data of those "outside" companies but purely the meta data hence you can balance security concerns with future sensing. You should use this meta data to identify new patterns that are suitable for provision as industrialised components (*B1* to *B2*). Once you've identified a future pattern then you should industrialise it to a discrete component service (*B3*) provided as utility and exposed through an API. You're now providing multiple components (*A2*, *B3*) in an ever growing platform of component services for others to build upon (*C1*). You then repeat this virtuous circle.

Obviously, companies in any space that you've just industrialised (*B2* to *B3*) might grumble — *"they've eaten our business model"* — so, you'll have to carefully balance acquisition with implementation. On the upside, the more component services you provide in your platform then the more attractive it becomes to others. You'll need to manage this ecosystem as a *gardener* encouraging new crops ("outside companies") to grow and being careful not to harvest too much. Do note, this creates an ever expanding platform in the sense of a loose gathering of discrete component services (e.g. storage, compute, database) which is distinct from a code execution platform (i.e. a framework in which you write code).

There is some subtle beauty in the ILC model. If we take our ecosystem to be the companies building on top of our discrete component services, then the larger the ecosystem is: -