complexity doesn’t go away simply by “outsourcing”. All you’ve done is try and pretend that the complex thing you’re building is somehow simple because then it’s easier to manage. It would be like BMW or Apple outsourcing their entire product lines to someone else and trying to have no involvement because it makes management simple.

Doctrine: Think small
In order to apply appropriate methods then you need to think small. You can’t treat the entire system as one thing but you need to break it into components. I will often extend this to using small contracts localized around specific components. Knowing the details helps you manage a landscape. But you can take this further and even use small teams such as cell based structures. Probably the best known approaches to using small teams are Amazon’s Two Pizza model and Haier’s Cell based structure.

Such teams should be given autonomy in their space and this can be achieved by the team providing well defined interfaces for others to consume along with defined boundaries often described through some form of fitness function i.e. the team has a goal around a specific area with defined metrics for delivery. Maps themselves can be useful in helping you identify not only the teams you should build but also the interfaces they need to create — see figure 40.

Figure 40— Think small (as in teams)