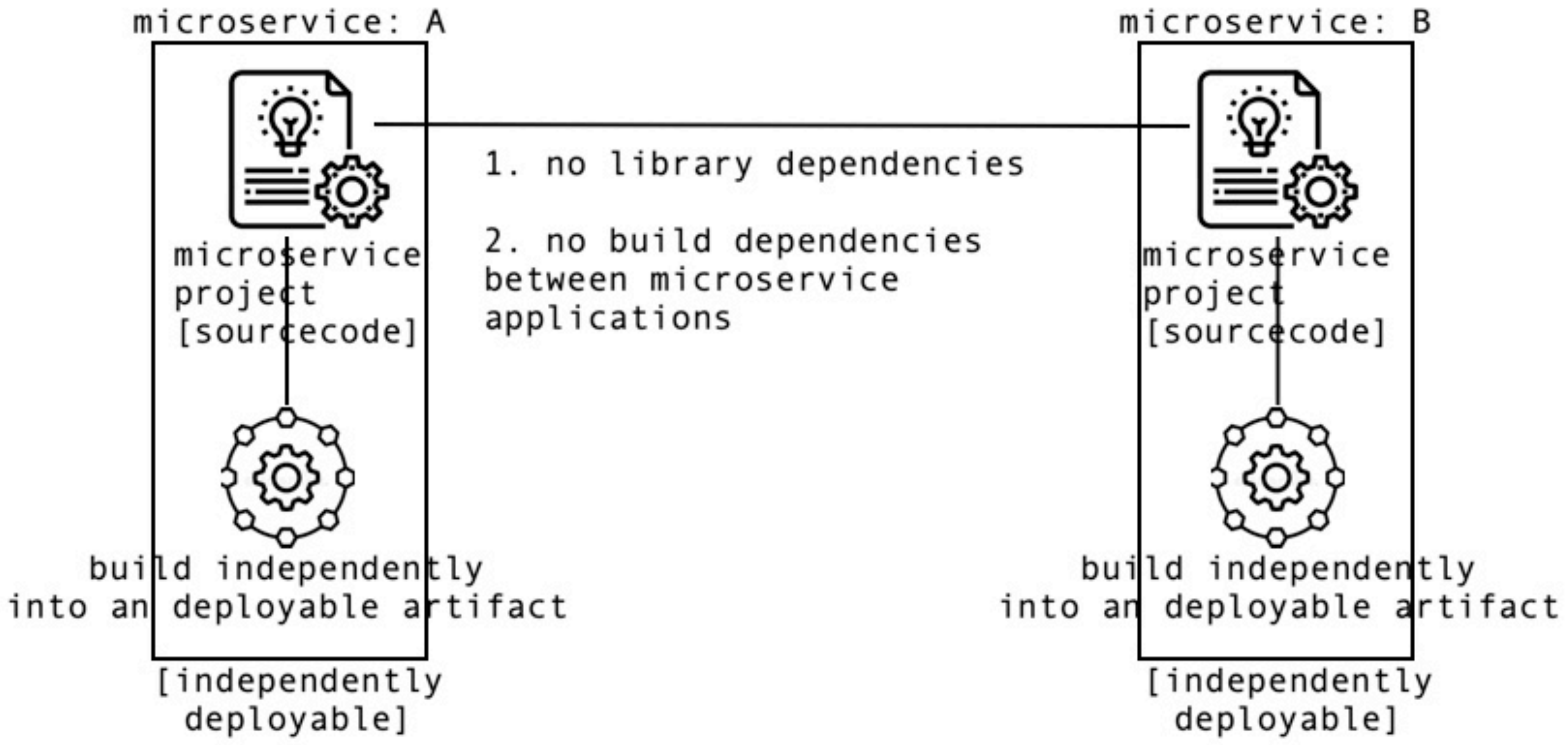

2. Independently Deployable: [Isolated from another Microservice completely]
These microservices should be independently deployable, isolated from another microservice. This means each microservice should be running out of its own container, multiple microservices should not be deployed on one-single container
There are lot of advantages of deploying them independently:
1. release management becomes easy. if we want to release latest version of one microservice application or patch an existing microservice application that is running, we can shutdown the existing container on which the microservice is running and can upgrade/patch and restart it independently without affecting the other microservices.

2. Incase of one microservice is crashed, it will not impact the other microservices as each of them are running on its own containers isolated from another.

3. The load on one microservice will not impact the others.

1. Loosely Coupled
Each microservice should be build/developed out of its own source-code project without having the other microservices as library/classpath dependencies. so that each of them will be loosely-coupled from others.

Microservice:A should be developed independently without referencing any of the classes of Microservice:B by addding the B as library/classpath dependency to the A.

There are advantages with this:
1. each microservice application can be developed by its own self-organizing indepenent teams that accelerates the development of the application
2. since no #2 components of #2 different microservices communicates with each other by holding interface references, a change in one microservice module will not affect the others, so that each team can plan and release their features without bothering the impact on other services.
3. all the members of the project dont need to have knowlege on all the modules of the system we are building, hence it makes the developers more focussed on the specific modules/microservice application there are working, that promotes quality development and reduces the complexity