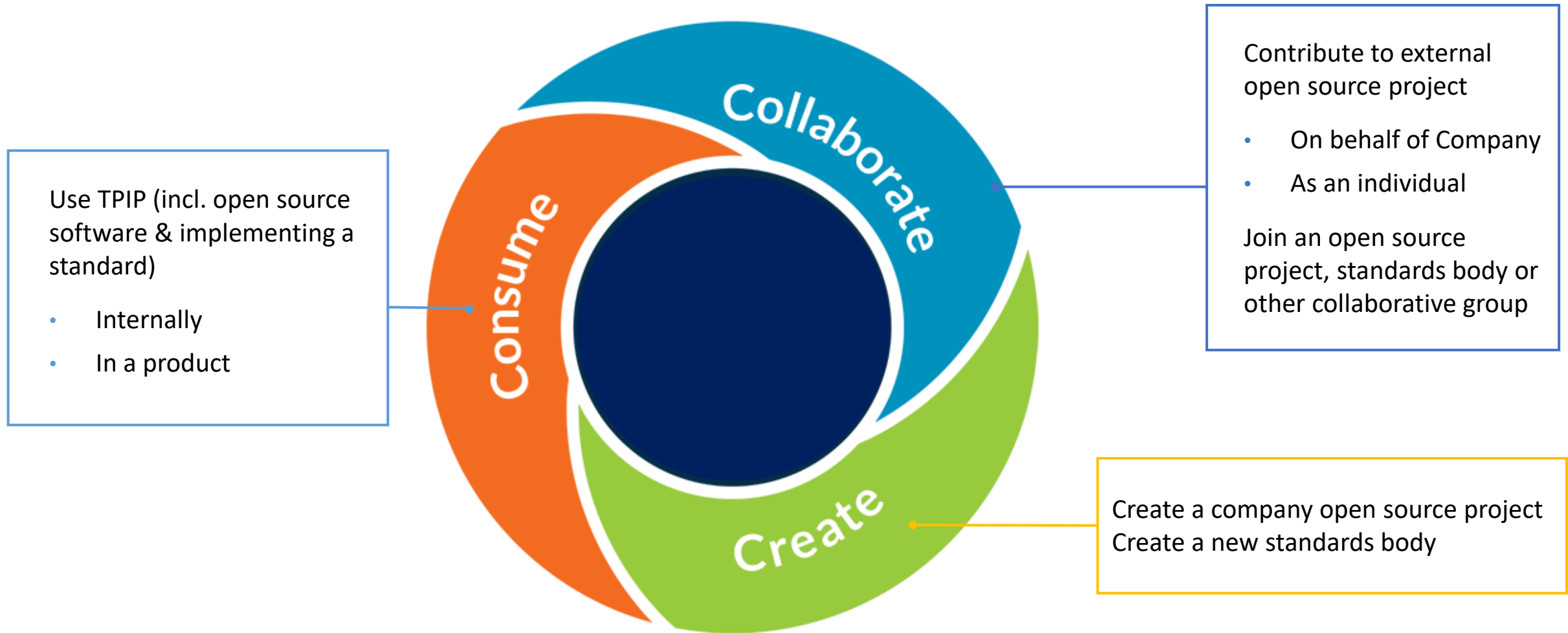


Future Open Source Leaders Training

Sami Atabani
Senior IP Consultant

How most organizations interact with Open Source



Types of Open Source Licenses

Permissive

- License requirements are minimal (e.g., retain notice; include copy of license)
- Broad grant of rights (with no conditions for particular licensing terms)

Copyleft

- Source code must be made available for binary distribution
- Original work, any modifications, any derivative work must remain under same license
- “reciprocal” or “hereditary”
- May apply to entire derivative work (strong copyleft) or only modified files (weak copyleft)

Types of Open Source Licenses

Permissive

- MIT
- BSD-3-Clause
- BSD-2-Clause
- Apache-1.1
- Apache-2.0

Copyleft

- MPL* (Mozilla Public License)
- CDDL* (Common Distribution and Development License)
- LGPL*
- GPL*
- AGPL*
- * = all versions

These are broad categories, specifics depend on the individual license.

Open Source Licenses: Projects and Files

- It is very common for (open source) software to include other open source software which may be under a different license.
- In this case, the “project level” license may not be the only license that applies.

Project Foo: GPL-2-only

Project A
BSD-2-Clause

Commercial Product: Restrictive EULA

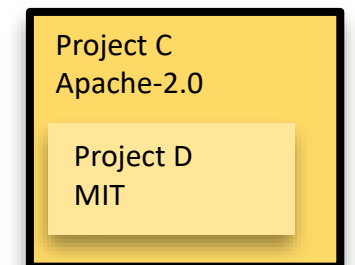
Project B
BSD-3-Clause

Project C
Apache-2.0

Project D
MIT

Multiple Licenses Apply

- Where multiple licenses apply, it is "conjunctive" and you have to comply with all the licenses.
- This is expressed with AND
 - Project Foo: GPL-2 .0-only AND BSD-2-Clause
 - Project C: Apache-2 .0 AND MIT



Multiple Licenses – Choice of License

- Where open source projects give you a choice of the license, it is "disjunctive" and you can choose which license to take the code under.
- This may be to enable the project to be combined with other projects under different licenses. It is expressed with OR:
 - MPL-1.1 OR GPL-2.0-or-later OR LGPL-2.1-or-later
 - Sometimes the license choice is between a commercial license and an open source license (usually GPL or AGPL).

```
<!-- The contents of this file  
are subject to the Mozilla  
Public License Version - 1.1  
(the "License") -->
```

```
<p> Alternatively, the contents  
of this file may be used under  
the terms of - either the GNU  
General Public License Version 2  
or later (the "GPL"), or - the  
GNU Lesser General Public  
License Version 2.1 or later  
(the "LGPL"). . </p>
```

Use Case: Distribution1

- The conditions in open source licenses trigger upon distribution.
 - Distribution = passing material to a “change of hands” (e.g., outside of company)
 - GPLv2 example: “Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted...”
- How the code is distributed makes a difference on the license conditions you must comply with:
 - Are you distributing in source or binary form?



Use Case: Distribution2

- Is access via a computer network a “distribution?”
 - Code used on the back-end or server side is not “distributed”
 - But, some licenses place conditions on access via a computer network
 - Examples: Affero GPL, Apple Public Source License, Reciprocal Public License
 - Code that downloads to the end-user’s computer is “distributed” and must be considered in terms of license compliance
 - Examples: Javascript, downloaded web client

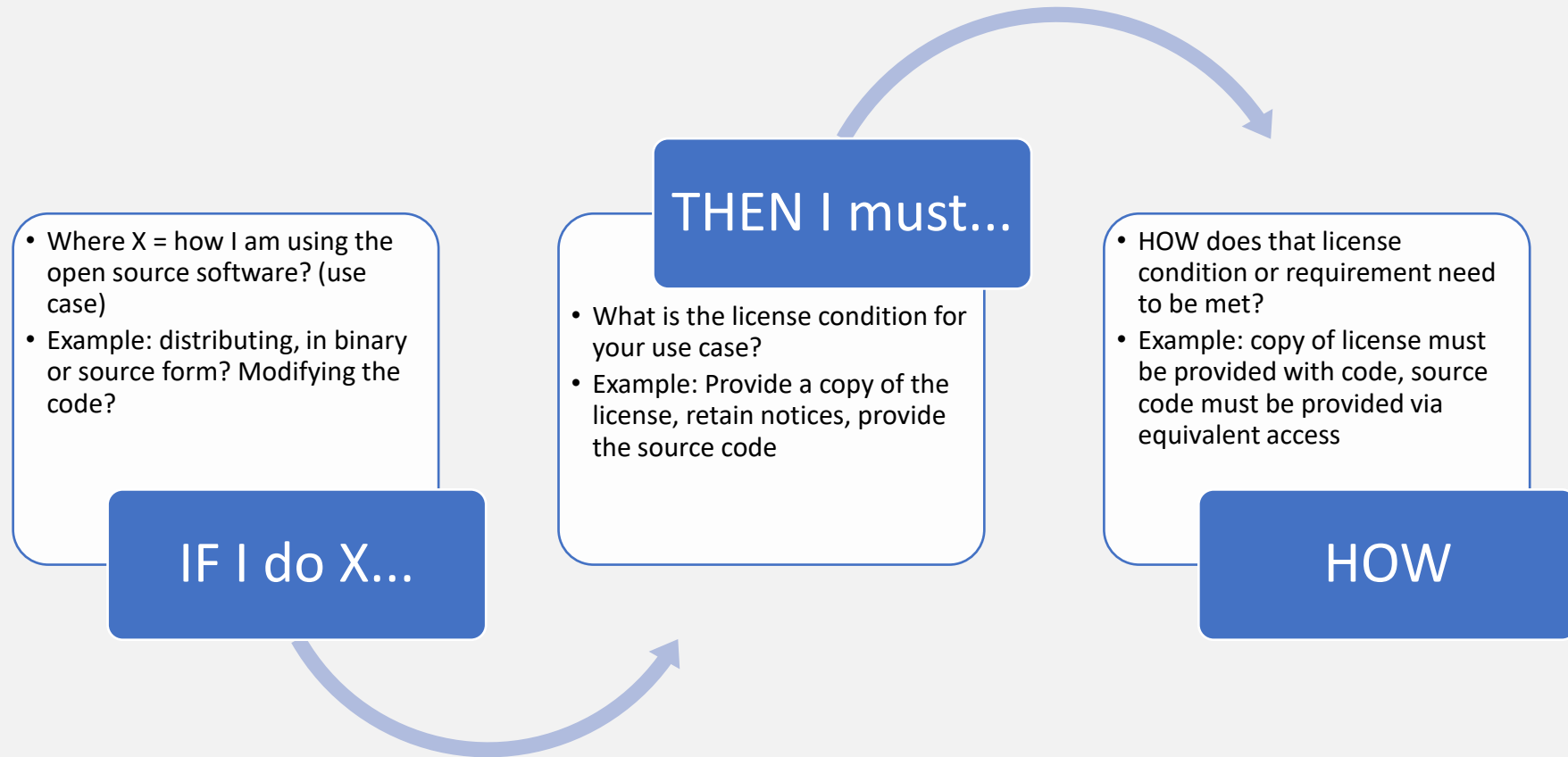


Use Case: Modifications

- Modifications include changes to the existing program
 - additions, deletions, new files, etc.
- Some open source licenses require notice of any modifications, so it's important to track this.
- Tracking modifications made is also considered good engineering practice!



Open Source License Compliance Analysis



Why do we need to manage Third Party IP (TPIP)?

- Risk analysis and management
 - Support and maintenance
 - License obligations and interaction with company agreements
- Business Implications
 - Potential impact on negotiations
 - Customers want to know!
- Improve efficiency
 - Repeat requests
 - Inconsistent processes lead to delay and frustrations



Managing TPIP needs to be proactive (not an exercise in putting out fires)!