



Proprietäre vs. offene Systeme

15. März 2021

Proprietär oder Open Source: Für die »echten« Computerexperten ist es ein Zwiespalt, für viele nach wie vor eine Glaubensfrage. Dieses Thema stimuliert vor allem bei Programmierern und ihren Artverwandten beizeiten angelegte Triggerpunkte.

Wer bereits in jungen Jahren als freiheitsliebender Linux-Nutzer der Marktmacht von Windows- und Mac-Betriebssystemen trotzte und dabei den innewohnenden Pioniergeist genoss, tut sich zunächst wohl auch im Beruflichen schwer, sich auf proprietäre Systeme einzulassen.

Proprietär: Das klingt nach Wohlstand und Abschottung, aber was genau ist darunter zu verstehen? In diese Kategorie fallen herstellerspezifische Soft- und Hardware, deren Benutzung durch den Urheber oder Rechteinhaber kontrolliert wird und dadurch in der Regel etwas kostet. Wie der Begriff vermittelt, ist es eine Frage der Eigentümerschaft, deren Wahrung und Vergütung grundsätzlich ja nicht verwerflich sind. Schließlich haben Programmierer und Entwickler eine Menge Arbeit und ihre Auftraggeber Geld investiert.

Das Gegenteil proprietärer sind offene Systeme, im Falle von Software also Freeware bzw. Open source – das sind die Programme, an denen jeder nach Lust und Laune »rumschrauben« darf. Dort ist der Quellcode frei verfügbar und das in der Regel kostenfrei. Der Großteil der Hobby-, aber auch der professionellen Programmierer bevorzugen sicherlich diese Variante, nicht nur aus Prinzip oder Kostengründen. Vor allem diejenigen, die gut vernetzt sind mit ihresgleichen, können ähnlich dem Wikipedia-Prinzip daraus ihr eigenes – natürlich offenes – Projekt machen.

Die Initiatoren wiederum, die den Quellcode erstellt und bewusst freigegeben haben, nutzen damit Know-how und Dynamik der Community und können mit den externen Weiterentwicklungen wiederum ihre Ziele weiterverfolgen. Gerade in Zeiten wie diesen, in denen Programmierer überall händierend gesucht werden, ist dies eine gangbare Form des Outsourcings. Klingt nach Win-Win-Situation, ist es oft auch.

Wenn allerdings die Aufgabenstellung wie etwa in der Automatisierungstechnik zu speziell formuliert ist, wird man mit einer Open-Source-Strategie nicht weit kommen. Somit bleibt der klassische Weg und der endet in der Regel in einem proprietären Produkt. Dazu zählen beispielsweise klassische SPS-Steuerungen von Siemens (Simatic S7), Kuka Knickarmroboter mit Steuerung oder Motion Control-Technik, also hochwertige Antriebstechnik bestehend aus Servomotor und Servoregler.

Doch auch in der Automatisierungstechnik tummeln sich neuerdings »echte« offene Systeme, zum Beispiel ctrlX Automation von Bosch Rexroth, PLC next von Phoenix oder EcoStruxure Automation Expert von Schneider Electric. Hier ist zwar nicht der komplette Quellcode für jedermann zugänglich, doch wurden »Türen« eingebaut, die Anbindungen an IoT oder die Cloud erlauben. Manchmal bezieht sich die Offenheit darauf, dass Open-Source-Software integriert ist. Auch dort, wo schon bislang IPCs statt klassischer SPSen verwendet wurden, gab es schon lange ein extrem größeres Maß an Offenheit, betont MGA-Chef Lorenz Arnold.

Für ihn ist der Zwiespalt zwischen proprietären und offenen Systemen ein alter Hut. Im Gegensatz zu bestimmten Nerd-Kreisen, in denen erstere mit negativen Eigenschaften wie konservativ, altbacken, geldgierig, den Nutzer entmündigend assoziiert oder als Innovationskiller bezeichnet werden, sieht er es anders: »In mir sträubt sich alles gegen diese pauschale Kategorisierung! Proprietär ist ja nicht grundsätzlich ein Nachteil für den Nutzer, ganz im Gegenteil: Geschlossene Systeme



MGA Ingenieurdienstleistungen GmbH
Die Automatisierungsexperten



bieten den grandiosen Vorteil, dass Teilsysteme ideal aufeinander abgestimmt entwickelt werden können. Wenn ich beispielsweise für eine Verpackungsmaschine im Hochleistungsbereich ein durchgehendes System von CPU, Servoregler, Antrieb bis hin zur Sicherheitstechnik verwende, so werde ich definitiv höhere Leistungen und Verfügbarkeiten erreichen bei einem überschaubaren Aufwand für die Applikation.«

Natürlich schätzt auch er die Vorteile offener Systeme: »Um ganz neue Lösungen zu entwickeln, um Wege zu gehen, an die die Hersteller der Automatisierungskomponenten noch gar nicht gedacht hatten, ist ein offenes System die einzige Chance.« Als Praxisbeispiel hat er ein aktuelles MGA-Eigenprojekt vor Augen: »Es ist beispielsweise so gut wie unmöglich, unsere Sprachsteuerung direkt auf der CPU eines proprietären Systems laufen zu lassen. Bei einem offenen System ist das wiederum ein Klacks!«

Dagegen nennt er auch ein Paradebeispiel für ein zweifellos extrem gelungenes proprietäres System, das iPhone: »Käme irgendjemand auf die Idee, dieses zu ›verdammen‹, nur weil es ein proprietäres System ist? Bewundern wir nicht alle – zumindest insgeheim – Steve Jobs als Visionär?«

Wobei die Visionäre von morgen auf zumindest halboffene Systeme setzen dürften: offen für KI – ohne Triggern.

← Zurück