

# Konsortiale Open Source Softwareentwicklung im Energiesektor

ew – Magazin für die Energiewirtschaft, Februar 2014

Preprint



Bildquelle: N-ERGIE Aktiengesellschaft.

Der Ausbau der Energie- und Wassernetze, neue Steuerungskonzepte für eine verstärkte Einspeisung von erneuerbaren Energien (EEG) sowie erhöhte Anforderungen an die IT-Sicherheit als Bestandteil der kritischen Infrastruktur sind einige der Themen, welche die Netzbetreiber mit ihrer technisch orientierten IT in naher Zukunft meistern müssen.

Dabei ist spürbar, dass es immer komplexer wird, neue Anforderungen in eine über Jahrzehnte gewachsene Systemlandschaft zu integrieren und die historisch gewachsenen Grenzen zwischen den einzelnen strategischen IT-Systemen zunehmend zum Hindernis werden. Dazu ist die Abhängigkeit von wenigen Herstellern für technisch orientierte IT-Systeme bei jeder Änderung oder Erweiterung zunehmend unflexibel. Zeitnahes Erfüllen der Anforderungen von Energiewende und Regulierung zu adäquaten Kosten wird damit zunehmend zum Problem.

Eine Gruppe von sechs deutschen Netzbetreibern hat sich 2012/13 entschlossen, innerhalb einer Machbarkeitsstudie [3] zu untersuchen, ob die gemeinschaftliche, konsortiale Entwicklung und Bereitstellung von Open Source Software für den Betrieb zukünftiger Stromnetze diese Situation entscheidend verbessern kann.

Der Empfehlung dieser Studie folgend, soll im Jahr 2014 ein Zusammenschluss interessierter Netzbetreiber, Hersteller, Dienstleister und Hochschulen in einem Umsetzungsprojekt den Beweis für die in der Studie gemachten Aussagen zur flexiblen und effizienten Softwareentwicklung anhand des Themas „Last- und Einspeisemanagement“ antreten. Dabei wird eine dynamische Entwicklung des Interessentenkreises nach Projektfortschritt angenommen. Ein positives Resultat ließe sich auch auf die anderen Netzsparten und weitere Netzbetreiberprozesse übertragen.

## Die IT-Systemlandschaft für den Betrieb von Netzen

Die IT-Systemlandschaft für den Betrieb von Netzen entstammt einer Zeit, in der die Rechnerleistung und die Speicherkapazität wesentlich geringer und damit teurer waren als heute. Um nach und nach einzelne Arbeitsprozesse durch Computer unterstützen zu lassen, mussten Einzelsysteme konzipiert werden, die Einzelaufgaben abarbeiten konnten.

Inzwischen haben sich die Möglichkeiten der technisch orientierten IT wesentlich weiterentwickelt, geblieben ist aber die Trennung in monolithische Einzelsysteme.

Die heutige Herausforderung besteht in zunehmender Verknüpfung dieser Systeme mittels standardisierter Schnittstellen, um so auf Daten in anderen Systemen zugreifen zu können, ohne Redundanzen zu erzeugen oder gar die Daten mehrfach manuell erfassen zu müssen. Dabei steigen die Komplexität der notwendigen Schnittstellen und damit auch die Kosten in erheblichem Maße. Dies führt wegen des zu erwartenden Aufwandes entweder zum Verzicht auf die notwendigen Verbindungen oder es muss in nichtfunktionalen und nichtproduktiven Datentransfer investiert werden.

Verschärft wird diese Situation durch die Bindung an einen Lieferanten pro System: Bei der Erstellung von konventionellen, proprietären Schnittstellen sind die Netzbetreiber immer auf die Hersteller der beiden betroffenen Systeme angewiesen. Es findet kein Wettbewerb statt, der geforderte Preis und der angebotene Termin muss vom Anwender akzeptiert werden.

Das trifft natürlich genauso auf notwendige Funktionserweiterungen eines der beteiligten IT-Systeme zu. Diese Funktionserweiterungen haben gerade jetzt, mitten in der Energiewende deutlich zugenommen: Neue Funktionen werden von verschiedenen Herstellern parallel entwickelt, wobei die Entwicklungskosten mehrfach anfallen und von den Netzbetreibern auch mehrfach bezahlt werden müssen.

Die Netzbetreiber sehen sich also aktuell Herstellern gegenüber, die, nachdem der Zuschlag für die Implementierung eines IT Systems erteilt wurde, in einer „Lock-in“-Situation, weitgehend konkurrenzlos über die Lebensdauer ihres Produkts den Netzbetreiber an sich binden. Ein Wechsel zu einem anderen Anbieter erzeugt schwer zu rechtfertigende Aufwände für die Datenmigration, neue Schnittstellen und verbessert die wettbewerbliche Situation letztlich nicht.

## Konsortiale Entwicklung von Open-Source-Software



Bild: Quelle OSB-Foundation (vorbehaltlich)

Die Entwicklung von Gemeinschafts-Open-Source-Software löst, nach Meinung der Studienteilnehmer die von proprietärer Software aufgeworfenen Probleme. Gemeinschafts-Open-Source-Software ist Software, welche einer Gemeinschaft, z. B. einem als Genossenschaft organisierten Konsortium gehört, aber gleichzeitig auf einer offenen Plattform unter einer Open-Source-Lizenz kostenlos bereitgestellt wird.

Die Bereitstellung der Software als gemeinschaftlich entwickelte Open-Source-Software kann zu breiter Beteiligung von Parteien, Unternehmen wie Privatpersonen, führen und damit die Software langfristig erheblich kostengünstiger, qualitativ hochwertiger, ohne unerwünschte Zwänge, sowie nachhaltiger und langlebiger machen [1].

### **Organisatorische Grundlagen**

Zur Erreichung dieser Ziele organisieren sich interessierte Unternehmen in einem Konsortium. Die Aufgabe eines solchen Konsortiums ist es, die notwendigen Prozes-

se zu etablieren und zu steuern, welche zur Entwicklung der Gemeinschafts-Open-Source-Software notwendig sind.

Das Konsortium schafft Rechtssicherheit für die beteiligten Unternehmen, in dem es als Eigentümer der Software und allem weiteren geistigen Eigentum, z. B. der Markenrechte, auftritt. Das Konsortium sichert sich dabei die notwendigen Eigentumsrechte, indem es die Softwareentwicklung als Aufträge vergibt und steuert. Jegliche Beiträge zum Softwareprojekt, ob bezahlt oder unbezahlt, können nur mit Rechteübertragung an das Konsortium geschehen. In der Umkehrung verspricht das Konsortium, die Software unter einer geeigneten Open-Source-Lizenz dem Markt zur Verfügung zu stellen.-

Die vom Konsortium in Auftrag gegebene Software wird nach den Prinzipien von Gemeinschafts-Open-Source-Software öffentlich und transparent entwickelt. Diese Form der Softwareentwicklung stellt sicher, dass kein beauftragtes Unternehmen Wissen an sich binden kann, welches das Konsortium zwingt, immer genau zu diesem Unternehmen zurückzukehren. Dies verändert die Selbstbestimmungsrechte weg vom Lieferanten- hin zum Kundenmarkt.

Vielmehr erlaubt dieser Prozess und die resultierende öffentliche Dokumentation aller relevanten Informationen, dass das Konsortium unterschiedliche Aufträge an unterschiedliche Lieferanten vergeben kann und damit ein System (im Open-Source-Software –Sprachgebrauch, Ökosystem genannt von Drittanbietern schafft, welche sich alle mit der Software auskennen und weitergehende Dienstleistungen, z. B. Betrieb und Wartung, anbieten können. Das Ökosystem hat geringe Eintrittsbarrieren, so dass unterschiedliche Unternehmen, auch kleine, als Anbieter am Markt auftreten können. Auf diese Weise werden die bekannten Probleme wie Vendor-Lock-in, mangelnde Ressourcenkapazität, Terminprobleme und zu geringe Innovationsgeschwindigkeit vermieden.

Die unterstützende organisatorische Struktur eines Konsortiums gibt es schon länger im Open Source Bereich. So hat sich z. B. die Luftfahr- und Verteidigungsbranche im Anwenderkonsortium TOPCASED zusammengefunden und entwickelt gemeinsam Open-Source-Software, um sich von der Abhängigkeit von einzelnen Werkzeugherstellern zu befreien (welche häufig als Unternehmen eine deutlich kürzere Lebenszeit haben, als die Flugzeuge, die sie mit Software unterstützen sollen). Die Luftfahrtbranche stellt hierbei einen guten Vergleichspunkt dar: Die Software ist langlebig und vor allem auch sicherheitskritisch, was auch auf die Software für die deutschen Stromnetze zutrifft.

Ein weiteres erfolgreiches Beispiel ist AUTOSAR, ein Konsortium von Automobilherstellern, welche sich zusammengeschlossen haben, um Softwarestandards, -modelle, und wieder verwendbare Komponenten für die Fahrzeugelektronik im Automobilbereich zu entwickeln. Auch hier handelt es sich um langlebige und sicherheitskriti-

sche Systeme, welche durch ein offenes und transparentes Vorgehen unterstützt werden. Erwähnt werden soll auch das sich gerade im Aufbau befindende openETCS Projekt der Bahnbranche.

## **Technische Grundlagen**

Der neuartige Lösungsansatz umfasst aus Sicht der Informationstechnik zwei Aspekte: (a) die Softwarearchitektur und -plattform und (b) den Entwicklungsprozess und die dazugehörigen Werkzeuge.

Die geplante Softwarearchitektur wurde grob in der oben erwähnten Studie skizziert und soll im Rahmen eines Umsetzungsprojektes konkretisiert und umgesetzt werden. Der sogenannte SOA-Ansatz ermöglicht die Anpassbarkeit und Integration in die existierende IT-Landschaft. Die Softwarearchitektur basiert auf zwei grundlegenden Prinzipien:

- Service-Oriented Architecture (SOA). Es wird eine dienstbasierte Architektur angestrebt, welche zur Modularisierung der Software dient. Diese Modularisierung hat folgende Vorteile: (1) Komponenten sind hinter ihren Schnittstellen austauschbar, so dass immer die beste Wahl getroffen werden kann, (2) Komponenten können unabhängig voneinander entwickelt werden, so dass Innovations- und Entwicklungsgeschwindigkeit dank Entkopplung groß bleiben kann, und (3) existierende Altsoftware kann als Dienst einfacher integriert und deshalb länger genutzt werden.
- Gemeinschaftliche Software-Entwicklung über die Bereitstellung von organisatorischen und technischen Plattformen. Statt neu anzufangen, wird unter Verwendung der SOA existierende monolithische Software inkrementell aufgebrochen und als getrennte Komponenten integriert. Neben den neuen fachlichen Anforderungen (Last- und Einspeisemanagement – siehe unten), welche neue Funktionen verlangen, können also auch bestehende Funktionen integriert werden, so dass die entwickelte Software jederzeit konkret werthaltig für die beteiligten Parteien ist.

Der geplante Softwareentwicklungsprozess und die dazugehörigen Werkzeuge wurden ebenfalls in der Studie betrachtet und sollen zusätzlich im Rahmen eines Förderprojekts vollständig definiert und umgesetzt werden.

Das wesentliche Merkmal des Entwicklungsprozesses ist, dass er transparent stattfindet und dass alle relevanten Projektartefakte für den Markt einsehbar sind. Diese höchstmögliche Transparenz gestattet breite Streuung von Wissen und verhindert den „Lock-in“ an bestimmte Anbieter. Eingebaut wird in den Prozess nicht nur Transparenz sondern eine entsprechende Qualitätssicherung, wie nur Open-Source-Softwareentwicklung sie erreichen kann.

Öffentlich und transparent entwickelte Software ermöglicht durch Peer-Review und Viele-Augen-Prinzip, die Software schneller fehlerfreier, stabiler, und sicherer zu entwickeln. Das Prinzip "Security through Obscurity" ist unter Fachexperten schon lange als nicht zutreffend diskreditiert worden. Der vermeintlichen Schwäche, dass die Software einsehbar ist und Sicherheitslücken schneller gefunden werden können, ist entgegenzuhalten, dass sehr viel mehr Personen auf der Suche nach Sicherheitslücken sind, um sie zu verhindern, als um sie auszunutzen.

Wie die Forschung von Prof. Schryen von der Universität Regensburg zeigt, ist die erhöhte Sicherheit von Software auf die Eigenarten des gewählten Softwareentwicklungsprozesses zurückzuführen. [\[2\]](#).

## Wirtschaftliche und technische Erfolgsaussichten

Die Motivation potentieller Interessenten und Anwender, sich an der Entwicklung und Vermarktung von Open Source Software zu beteiligen, ist sicherlich damit verbunden, die wirtschaftlichen und technischen Chancen und Risiken dieser Idee einzuschätzen. Deshalb war es einer der wichtigen Schwerpunkte, innerhalb der Studie, dazu verlässliche Parameter für eben diese Einschätzung zu gewinnen.

Im Ergebnis wurden die wirtschaftlichen und technischen Erfolgsaussichten des vorgeschlagenen Projekts als sehr gut bewertet. Zwar ist Softwareentwicklung immer mit Risiken behaftet, es hat sich hier aber eine Gemeinschaft gefunden, welche alle notwendigen Kompetenzen, fachlich wie technisch, mitbringt. Da die Software direkt an den anstehenden Bedürfnissen der teilnehmenden Unternehmen entwickelt wird, erwarten wir einen starken Fokus auf kontinuierliche Werthaltigkeit (und entsprechende Steuerung).

Das vorgeschlagene Entwicklungsmodell von Gemeinschafts-Open-Source-Software mag ungewöhnlich erscheinen, ist es aber nicht. Open Source Software wird seit mehr als 20 Jahren gemeinschaftlich über Unternehmensgrenzen hinweg entwickelt. Dies bezieht sich seit längerem nicht nur auf Endkunden-Anwendungen sondern fand Eingang in industrielle Anwendungen. Es liegen also entsprechende erfolgreiche Erfahrungswerte und Kompetenzen aus anderen Branchen vor.

Das für die Software im Energiebereich zu gründende Konsortium wird eine Non-Profit-Organisation sein, die im Interesse ihrer Mitglieder sowie der Öffentlichkeit agiert. Die zu entwickelnde Software ist für die beteiligten Unternehmen nicht wettbewerbsdifferenzierend. Es handelt sich um gemeinschaftlich entwickeltes und besessenes Eigentum. Die Software reduziert die Kosten, verhindert Vendor-Lock-in, etc. stellt aber keines der beteiligten Unternehmen besser als ein anderes.

Gleichwohl ermöglicht das Konsortium durch die Wahl einer geeigneten Open-Source-Lizenz („unterschiedliche Ausprägungen des copyleft“) sowie unterstützende Prozesse die Erzeugung von Software und Dienstleistungen auf Basis der Gemeinschaftssoftware. Unternehmen können somit die Software verwenden, erweitern und Produkte und Dienstleistungen darauf aufsetzend verkaufen. Somit ergibt sich sehr wohl Wettbewerb und Innovation in einem von dem Konsortium ermöglichten Ökosystem an Softwarelieferanten und -dienstleistern. Gleichwohl bleibt das Konsortium, bei erfolgreicher Zusammenarbeit immer in der Initiativrolle und kann somit sicherstellen, dass keiner der verwertenden Anbieter so dominant wird, dass doch ein Lock-in stattfindet.

Das Konsortium ist offen, nach geregelten Bedingungen weitere Interessenten der Energiebranche aufzunehmen wie auch Softwareunternehmen und weitere Institutionen.

[1] <http://dirkriehle.com/2010/01/06/the-economic-case-for-open-source-foundations/>  
Dirk Riehle. "The Economic Case for Open Source Foundations." IEEE Computer, vol. 43, no. 1 (January 2010). Page 86-90.

[2] Schryen, G., Is Open-Source security a myth? What do vulnerability and patch data say?, Communications of the ACM, Vol. 54, No. 5, pp. 130-139, 2011.

[3] Verlinkung auf das Open DOC der Studie. Stand Oktober 13