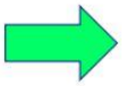
# MilramX Overview

Deep Six your Wasted Time with a MilramX based
DSIX (Decision Support and Information Exchange) System



## Introduction

As managers, we all spend too much time on unproductive meetings, too much paperwork, duplicate data entry, reading through reports, staring at computer screens, and “walking the floor” to try to figure out what is going on, or, more to the point, what is going wrong, so we can take corrective action, before it is too late.

Nowhere is this truer than in industrial enterprises.

MilramX is a DSIX software platform which enables the rapid implementation of real-time decision support and automated information exchange systems within industrial enterprises.

The purpose of these DSIX systems is to make sure that everyone within the organization has the information they need to do their job, when they need it, without drowning in paperwork or Excel spreadsheets.

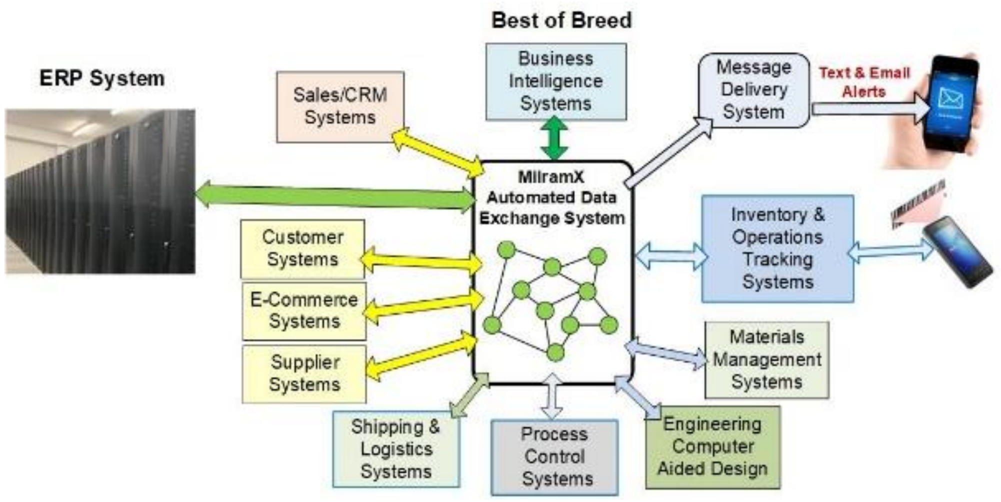
MilramX does this by automating the collection of new or updated data from a variety of systems, translates this data into information in a format usable by its recipients, and then forwards the information to systems used by the recipients and/or sends email or text messages to the recipient people who need this information.

## Important Applications

### *FDA Regulated Supply Chain Materials Tracking & Traceability*

An important use of MilramX is in FDA regulated supply chain applications, such as food, pharmaceuticals, medical devices and suplies, and cosmetics where it is used to exchange materials tracking and traceability data between trading partners. In such applications, MilramX can automatically exchange data in the form of EPCIS (Electronic Product Code Information Services) and EDI (Electronic Data Interchange) files  as well as provide the data needed for informational websites.

### *ERP System Augmentation*

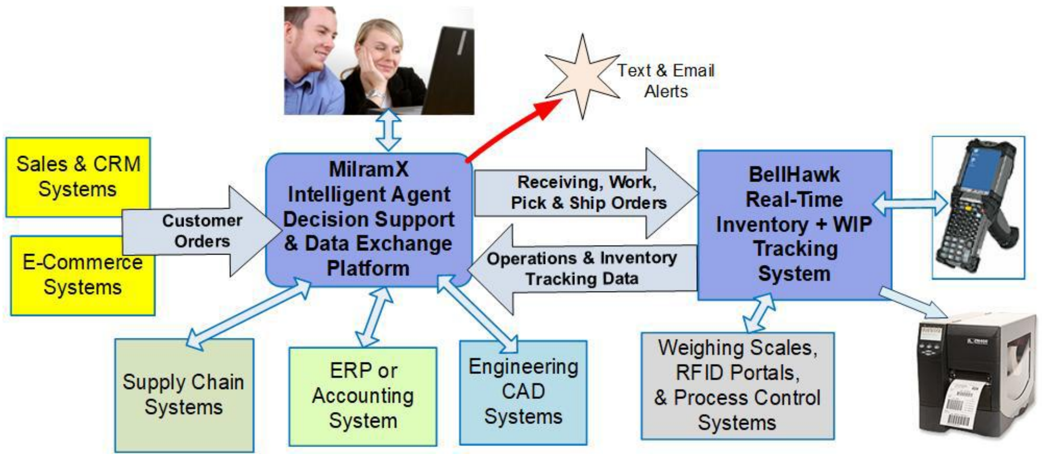

Best of Breed
ERP System
Sales/CRM Systems
Business Intelligence Systems
Message Delivery System
Text & Email Alerts
MilramX Automated Data Exchange System
Customer Systems
E-Commerce Systems
Supplier Systems
Inventory & Operations Tracking Systems
Materials Management Systems
Shipping & Logistics Systems
Process Control Systems
Engineering Computer Aided Design

An important use of MilramX is to enable the augmentation of existing ERP systems by exchanging information with other systems, typically over the Internet.

This includes automating the translation of customer orders, from a variety of sources, into work, picking, packing and shipment orders, for use within the ERP system, as well as the generation of purchase requisitions to purchase the needed materials. It also including exchanging information with specialized inventory and work-in-process tracking systems such as the BellHawk software to track warehouse and manufacturing operations.

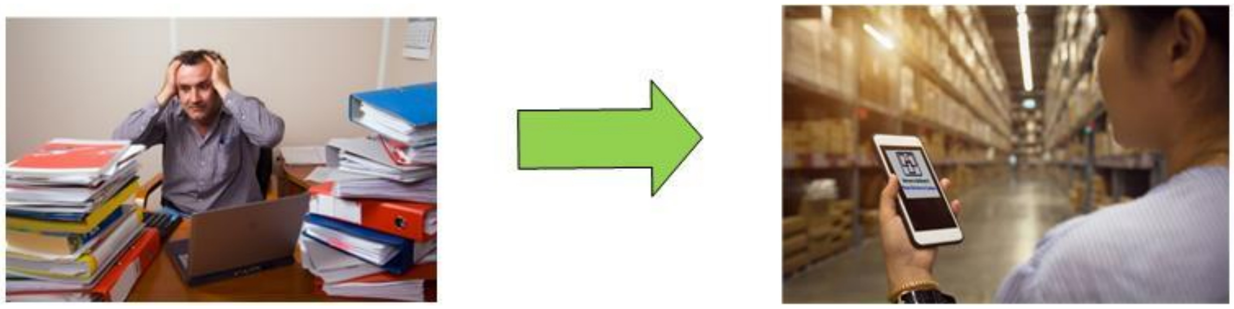
Note that in MilramX, we distinguish between data collected from source systems and information, which is what people need to do their jobs efficiently. This information is derived from the data, usually by Python scripts, and is then sent to systems used by the people who need the information in near real-time.

### *Adding Barcode & RFID Data Capture Capabilities*


Text & Email Alerts
Sales & CRM Systems
E-Commerce Systems
Customer Orders
MilramX Intelligent Agent Decision Support & Data Exchange Platform
Receiving, Work, Pick & Ship Orders
Operations & Inventory Tracking Data
BellHawk Real-Time Inventory + WIP Tracking System
Supply Chain Systems
ERP or Accounting System
Engineering CAD Systems
Weighing Scales, RFID Portals, & Process Control Systems

MilramX can be used to add barcode & RFID based inventory and WIP tracking capabilities to ERP and other systems through its interface to the BellHawk data collection software. BellHawk also includes the capability to automatically print barcode labels on demand and through MilramX to interface to process control lines and test stands.

## MilramX Concepts



1. **Real-Time Information:** ERP systems can provide great reports about what happened yesterday but generally do not provide the decision support information, which managers and their staff, as well as customer support and sales people, need in real-time to do their jobs. The focus of MilramX is providing the needed information in near real-time by frequently examining many different sources of data for changes and then converting these into actionable decision support information for many different people.

2. **Information and not Data:** What people need is usable information on which they can make decisions and not simply data moved from one system to another. In MilramX, this is accomplished by being able to use Python scripts to translate data into actionable information.

3. **Eliminate Duplicate Data Entry:** By automating the translation and movement of data from source systems to target systems, a large amount of unnecessary time spent on duplicate data entry into multiple systems, as well as the resultant mistakes, can be avoided.

4. **Eliminate the use of Paperwork:** Organizations tend to use paper forms and emails as a means of communications, when systems do not talk to each other. MilramX can eliminate most of this paperwork by automating information exchange between systems.

5. **Break Down Information Silos:** One of the great sources of inefficiency and customer unhappiness within organizations is due to a lack of information sharing between different groups of people and departments within the enterprise. MilramX can overcome this by automating the exchange of information.

6. **Eliminate the Use of Excel Spreadsheets.** Many organizations make extensive use of Excel spreadsheets to extract data from systems and then to manually analyze this data into actionable decision support information. Through is use of Python scripts, MilramX is able to automatically convert data to actionable decision support information, without first exporting the data to an Excel spreadsheet, and then manually manipulating the spreadsheet data.

7. **Many things happen in Parallel:** In a typical enterprise, there are multiple sources of new and updated data and many people who need the resultant data. MilramX supports multiple transfers of information happening at the same time. This includes retrieving many different types of data object, from multiple systems, converting this data to information, and delivering the resultant information to multiple other systems, all in parallel.

8. **Ease of Customization** ERP systems typically have very limited ability to be customized, as they need to provide a common body of software to many thousands of users. Also, development of data exchange interfaces to other systems typically requires the services of an experienced full-stack computer programmer. With MilramX the data exchange and information translation scripts are written in Python, which can readily be modified by business analysts, IT staff, and engineers with limited programming skills.

9. **Reuse of Code:** All ERP systems, as well as many other systems used by industrial organizations now have readily available Python code libraries for exchanging data with these systems. MilramX makes it straight forward to use these libraries in implementing an enterprise-wide decision support and automated information exchange system.

10. **Data Exchange:** It is now almost mandatory to exchange data, in a wide variety of formats, with supply chain trading partners, such as customers and vendors. Doing this using manual data entry can consume a large amount of labor time and cause expensive mistakes. MilramX automates these data exchanges when new or updated data is detected. It also reduces the complexity of customizing these data exchanges to writing or modifying Python scripts, with all the complexity embedded in readily available libraries.

At the core of MilramX is the concept of many intelligent agents running in parallel 24x7 to carry out "intelligent grunt" work" for people instead of having managers or their staff do tasks that are repetitive, labor-intensive, or time-consuming but still require some level of skill or knowledge to be done effectively.

Regenerative AI algorithms, are based on statistical correlation and most of the time produce a good answer but sometimes do not. By contrast, MilramX intelligent agents typically use rules, mathematical calculations, decision trees, and other algorithms which always produce deterministic answers that can be relied upon.

This does not mean that MilramX agents cannot use non-linear adaptive correlators (on which regenerative AI is based) for image recognition and the like but the decisions based on their output are always deterministic.

Like people, MilramX agents can also iterate their way, over time, to better information and recommendations, as more data becomes available. This is unlike regenerative AI algorithms, which produce a one-time answer, which may be right or wrong.

## Why use a DSIX system?

Twenty years ago, one of our manufacturing clients had about 100 people working in production and the warehouse with 20 people in the front office. When we last visited them, they had 6 people in production, running highly automated machines to make inserted fibers for cables, with 4 people in the warehouse, and 32 people in the front office.

The cost of their expensive overhead staff caused this otherwise very profitable operation to lose money and, as a result, they sold out to a much larger organization.

When we visited yet another company, a contract manufacturer of electromechanical systems, we spent the morning collecting copies of different paper forms that they were using to run their business, which is always a great way to start implementing a DSIX system.

This business employed about 30 people in manufacturing and the stock room, with about 14 people in the front office. At a meeting, after lunch, we spread out all 84 paper forms, which they routinely used to run their business, on their conference room table, to the amazement of their President and the members of his senior management team.

Both organizations were using relatively modern ERP systems and, in fact, the latter organization was using three ERP systems, with the main system costing over $40,000 per year.

So, what were these front-office people spending their time doing?

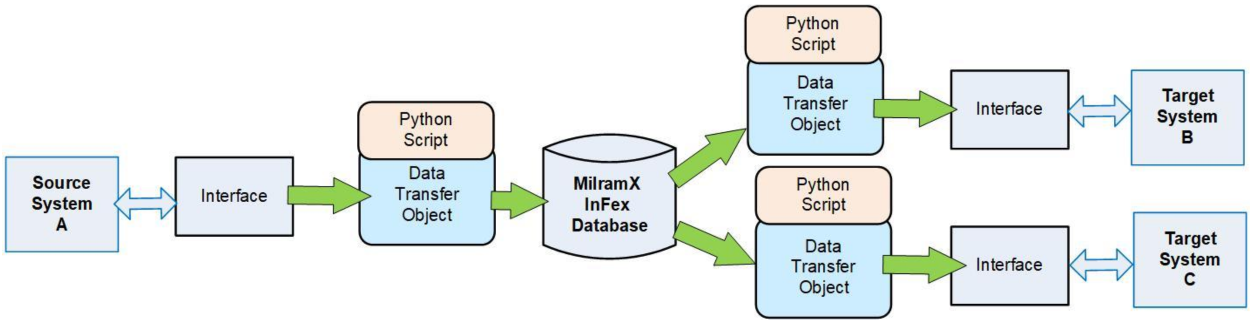
1. Attending coordination and meetings.
2. Talking to customers, suppliers, or employees to exchange information
3. Filling out paper forms to send to other departments
4. Entering data from paper forms into their ERP systems
5. Entering duplicate data in multiple systems
6. Reading reports, paper forms, or data on computer screens to try to detect operational problems
7. Walking-the-floor to try to find out what is going on and to head-off problems
8. Checking levels of available inventory, planning materials purchases, and scheduling production
9. Exporting Excel spreadsheets from their ERP or other systems to perform analyses
10. Exchanging data with customers and suppliers by email or other electronic means.

While face-to-face meetings can be important, most of the other tasks can be automated with a real-time decision support and automated information exchange system, such as MilramX; and even the need for many coordination and planning meetings can be significantly reduced.

Today, just like with our Federal Government, we are seeing many of these activities in industrial organizations resulting in substantial and unnecessary overhead costs which can be minimized with the use of a system like MilramX.

## Technical Overview

Conceptually, MilramX provides a framework to automatically schedule and run multiple intelligent-agent processes, called Data Transfer Objects (DTOs), in parallel. The function of some DTOs is to periodically collect the latest updates to a specific class of business objects, such as customer orders, from a specific source system, such as an ERP system, and then deposit these in the MilramX Information Exchange Database (InFex).



The function of yet other DTOs is to collect the updates to the InFex, interpret these, such as converting customer orders into picking and shipping orders, and to deliver these to appropriate target systems.

The purpose of the InFex database is to ensure that information can be delivered reliably between systems and also to enable information from one system to be delivered, in different formats, to multiple other systems.

Typically, data is exchanged between the ERP system and its auxiliary systems through web-services interfaces, over the Internet. Unfortunately, the Internet is becoming increasingly unreliable due to occasionally becoming overloaded and having to drop messages, as well as from physical causes, such as down lines.

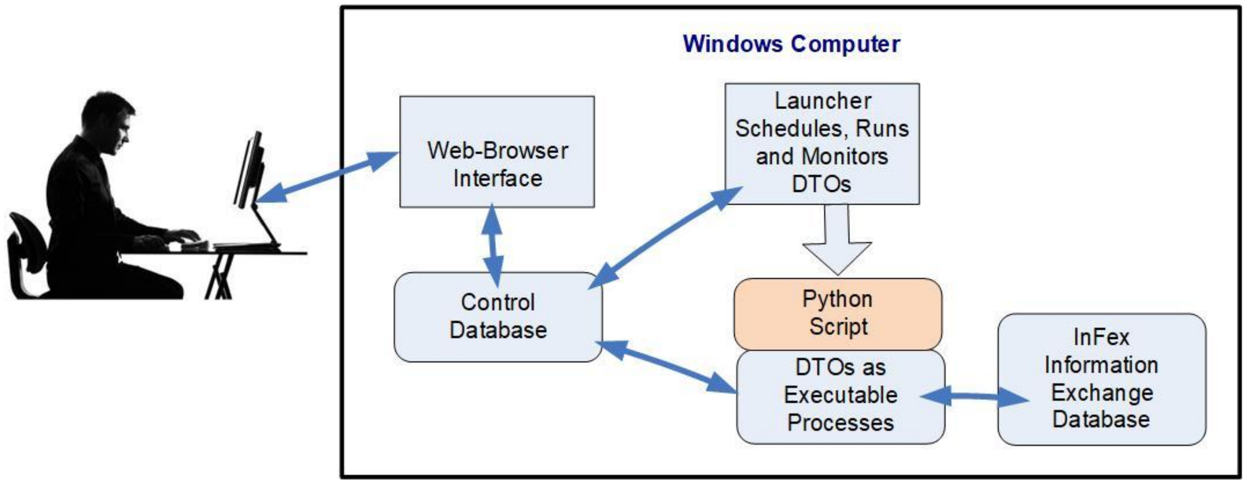
To help ensure that data is delivered reliably, data is first delivered into the InFex, so that fetching the data can be retried in the event of failure. Then, sending the fetched data to the target system, can also be separately retried, by another DTO thereby substantially increasing the reliability of data exchange between systems.

In addition to these data delivery DTOs, there are other DTOs that handle the automated sending and receiving of files, Emails, and text messages.

DTOs are typically written as Python scripts that call upon support dynamic link libraries (DLLs), which do most of the work in reliably transferring data. But where needed, DTOs can also be written in .Net to perform more complex functions.

In most applications, the function of the Python scripts is to interpret data from one system and/or to put it into a form that is usable by other systems, or by people to whom the DTO is sending Emails or Text messages. In this way, considerable decision-support capabilities can be integrated into the DTOs. This is in contrast to most data exchange frameworks which simply exchange fixed data between systems.

MilramX can be run on a Windows Server, workstation, or IOT Enterprise based system.

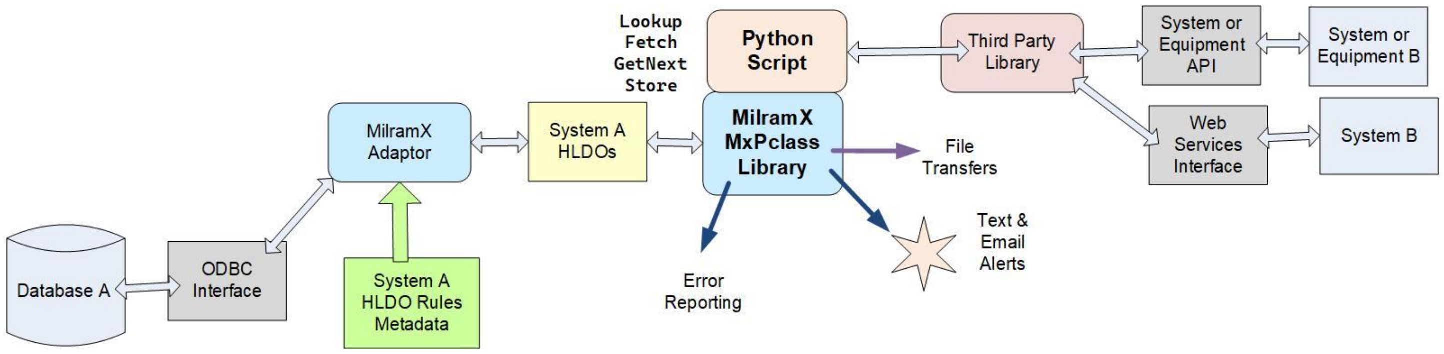

Windows Computer
Web-Browser Interface
Launcher Schedules, Runs and Monitors DTOs
Control Database
Python Script
DTOs as Executable Processes
InFex Information Exchange Database

It includes a web-browser interface through which DTOs can be scheduled and monitored, with all the scheduling data, as well as support services, handled through a Control database.

A system-level Launcher process, which runs continuously, then takes the data in the Control database and uses this to periodically launch DTOs to automatically look for updates in source systems and to deliver updates to target systems.

This then reduces most of the work of implementing real-time interfaces that run in real-time in parallel between multiple systems to that of writing Python scripts for DTOs to do the data interpretation.

## Use of Python Libraries



In MilramX, the Python scripts are just “the tip of the iceberg” and most of the work of exchanging data is done by support libraries which are typically written in .Net.

There are now Python libraries readily available for most ERP and many other systems used by industrial enterprises. These libraries can be installed be installed on the MilramX server and used by a DTO Python script to exchange data with a wide-variety of systems.

The MxPclass library, supplied with MilramX, enables DTOs to exchange data with databases, such as the InFex and Control databases. This library performs automatic translation between the complex tables, indirect references, and stored procedures of a typical database and the fetching and storing of data by the Python script of JSON strings.

This translation is performed using High Level Data Object (HLDO) metadata, for each database, which is stored on the server in the form of XML metadata which can be edited through the MilramX web interface, using Excel spreadsheets.

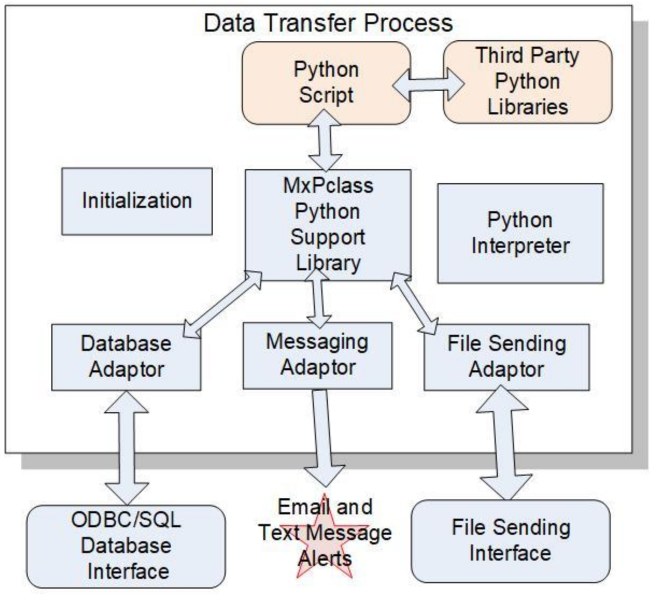
HLDO metadata is supplied for over 100 named data objects in the InFex, along with a copy of the InFex, database, as part of the installation package for MilramX. Both the InFex and its HLDO metadata can be customized as needed.

The MxPclass library also includes features to report errors and log these in the daily log file. It also includes features to support sending of Emails and, through third party libraries, text messages, as well as sending files to other systems using protocols such as SFTP and AS2.

## MilramX Data Transfer Objects

Data Transfer Objects (DTOs) are named executable objects run by the Launcher as separate Windows processes.

Associated with each DTO name is an executable .exe file to be run by the Laucher, to run the DTO. In MilramX, Python scripts are run by the Launcher as separate processes, using Python.exe as the executable process to which the script is passed at time of process creation.



These Python scripts can then call upon the MxPclass library, supplied with MilramX, to access databases, using high level methods such as Lookup, Fetch, Get Next, Store to translate between JSON strings and the reading and writing of data objects in the database.

The MilramX initialization file, which is installed as part of the MilramX IIS based web-browser interface, contains named Adaptor sections which define the access parameters for each disk or web-services interface.

Python scripts can also use the MxPclass library to retrieve the login access information for web-services interfaces from the MilramX initialization file.

The translation between JSON strings used to Fetch or Store database data and the table structures and/or stored procedures are controlled by High Level Data Object (HLDO) metadata.

Standard metadata files are provided with MilramX, such as for the InFex, and can be edited if needed by system implementors.

To support transferring just the latest updates, MilramX provides support for tracking time-tagged data and for just transferring new or latest updates.

A typical script for retrieving a class of data from an ERP system and writing it to the InFex is as follows:

1. Use 3rd party library to initialize ERP interface
2. Use MxPclass to initialize access to the control and InFex databases.
3. Fetch latest updates to data objects from ERP system
4. Store updates in InFex as then are retrieved
5. Close interface
6. Kill process

The DTO script for reading from the InFex and writing to an ERP system is as follows:

1. Use 3rd party library to initialize ERP interface
2. Use MxPclass to initialize access to the control and InFex databases.
3. Fetch latest updates to data objects from ERD
4. Process updates sequentially using GetNext and then use third party library to output to ERP system.
5. Close interface
6. Kill process

Note that each DTO transfer process is a self-contained process that links in DLLs as needed. It initializes the interfaces each time and kills itself when completed, thereby freeing up all the system resources that it consumed. Also, if it runs too long, due to a "hung" interface, the launcher can kill the process thereby freeing up systems resources and try again later.

The MxPclass Store method tracks when the store takes place and allows for duplicate receipts of the same data, in the event of the need to repeat a transfer. MilramX also tracks the time that each DTO last read a specific class of data objects from the InFex and only updates the time if the send is successful. This allows for repeating tries when data sent to a web-services interface that was unsuccessful.

While the primary function of DTOs is to transfer data, the Python scripts can also be used to provide decision support services, such as automatically converting an incoming stream of customer orders into picking and shipping orders for a warehouse.

## 24x7 Operation

MilramX is designed to run 24x7 for months-on-end of continuous operation. This is because there may be dozens or even hundreds of DTOs running to support the data exchange requirements of different departments within a busy enterprise.

To this end, the Launcher monitors processes that it has launched to make sure that they do not run too long. This can occur when a call to a third-party-library hangs (typically due to a hard failure in the middle of a transfer) or a database query takes way too long, maybe due to a circular reference.

If this happens, then the Launcher kills the process and will retry a specified number of times, before suspending the DTO process and sending an Email to one or more IT support people, so that they can come in remotely and fix the problem.

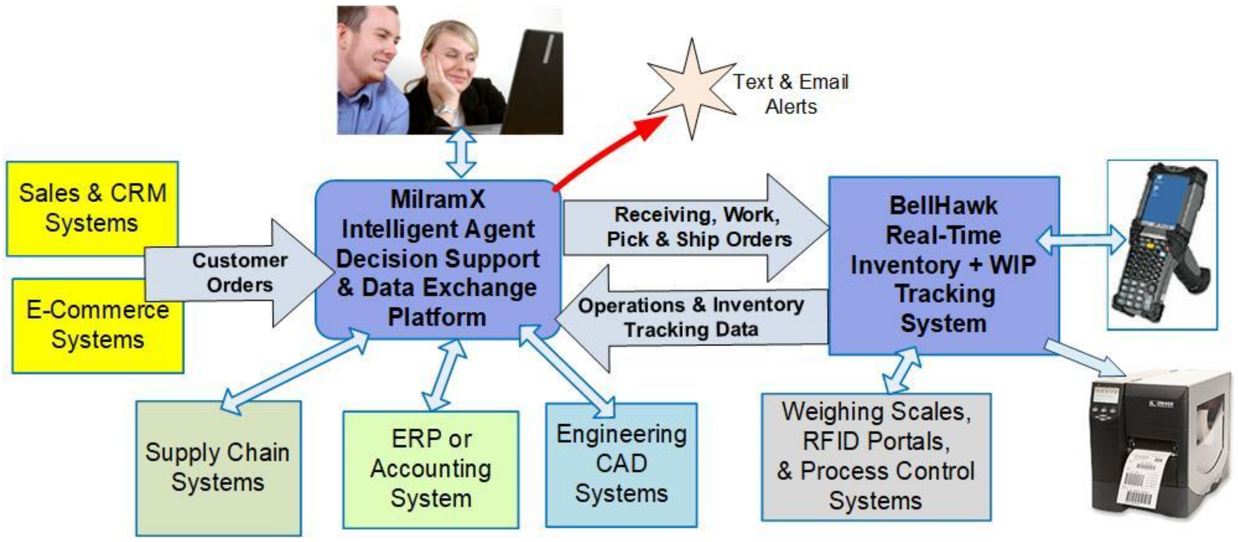
If bad data is detected in a transfer, the DTO can send an Email to IT support, and then free itself, thereby freeing up resources.

This killing of processes and freeing of DTO resources at the end of each run is important to long run operation, otherwise the computer on which MilramX runs would quickly run out of resources and stop running.

As Python scripts link dynamically with their supporting libraries, new or replacement scripts can be uploaded while the system is running other DTOs. There is no need to compile, link and upload a new data transfer program whenever a change has to be made to a data transfer or a data transfer added.

New DTOs can be added to a running MilramX through its web-browser interface, including uploading the Python script for the DTO, after installing any needed third-party interface libraries. Again, this enables MilramX data transfers to be dynamically expanded without interfering with ongoing data transfers.

## MilramX and BellHawk


Text & Email Alerts
Sales & CRM Systems
E-Commerce Systems
Customer Orders
MilramX Intelligent Agent Decision Support & Data Exchange Platform
Receiving, Work, Pick & Ship Orders
Operations & Inventory Tracking Data
BellHawk Real-Time Inventory + WIP Tracking System
Supply Chain Systems
ERP or Accounting System
Engineering CAD Systems
Weighing Scales, RFID Portals, & Process Control Systems

MilramX is often used in conjunction with the BellHawk real-time inventory and work-in-process tracking system. This is because ERP systems often lack the capability to track individual serialized items, or containers of material with different lot numbers and expiration dates, or rolls, reels, and sheets of material with different dimensions. They also often have trouble tracking totes of WIP materials for different customer orders as they flow from work-center to work center, as well as to track the shipment of mixed pallets.

To support such applications, there is a standard set of DTOs available that can exchange data between the MilramX InFex and the BellHawk database through the BellHawk SOAP/XML web-services interface,

### Commentary

The problem with implementing data exchange interfaces is not in reading data from, or writing data to, a database or web-services interface.

Rather, all the complexity comes in ensuring that updates to source system data are translated correctly and the resultant information delivered reliably to target systems, at the same time many other transfers are occurring. Plus, even more complexity is required to handle the bad data and failed communication errors that inevitably occur.

By handling all this complexity, MilramX eliminates the need for over 90% of the code that would otherwise need to be developed for these interfaces. All that is left is entering some setup data and writing the organization specific Python code scripts. And, even here, Python script development is simplified by the data abstractions supported by MilramX.

As a result, creating and modifying these scripts is something that can be done by business analysts, IT people, and engineers with limited programming skills. This is, as opposed to writing data exchange interfaces from scratch, which typically require an expensive full-stack .Net programmer with many years of experience.