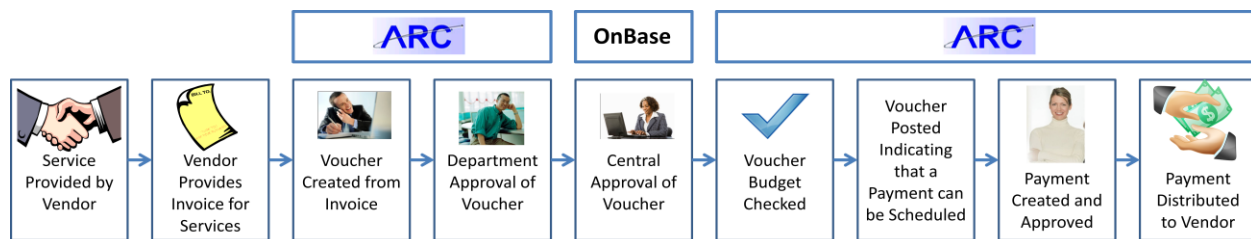


Accounts Payable End-To-End Process Overview

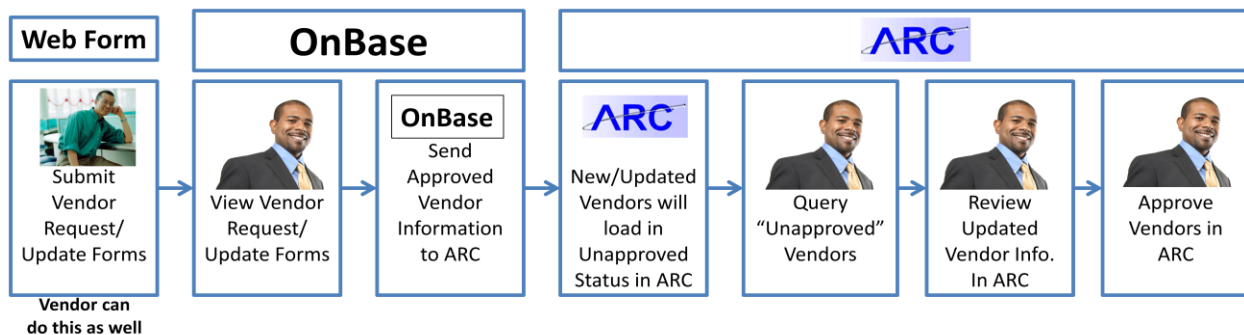
Introduction to Accounts Payable

The end-to-end Accounts Payable process encompasses all the activities that occur from requesting a new vendor in the ARC system through that vendor receiving payments for goods and/or services provided to Columbia University. The following job aid outlines the overall payment lifecycle and will help connect the multiple components that comprise ARC's Accounts Payable Module.

The high level Accounts Payable end-to-end process is as follows:



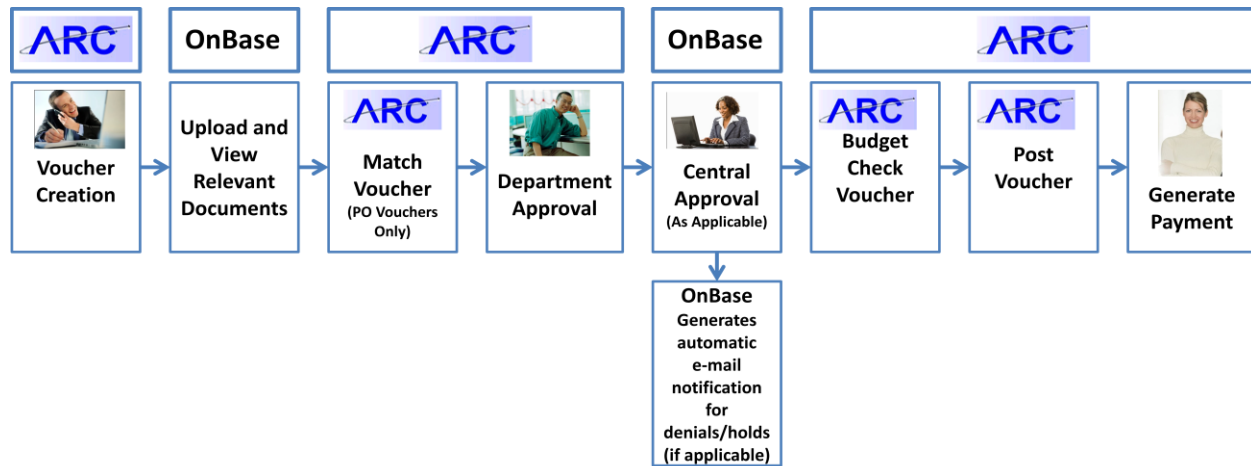
In order to received goods and/or services in ARC, the vendor first needs to be entered into the system. Vendors are requested via a vendor request web-form and entered into ARC by the Vendor Management Team:



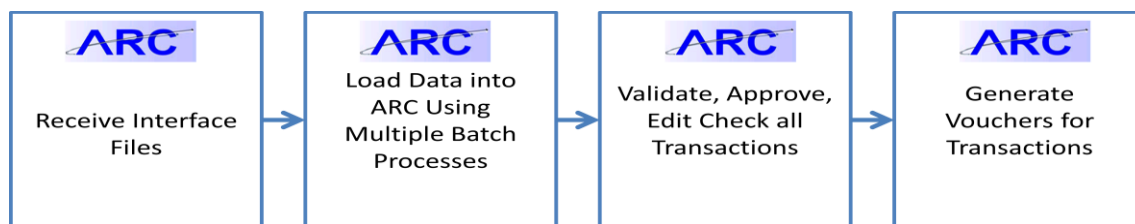
JOB AID



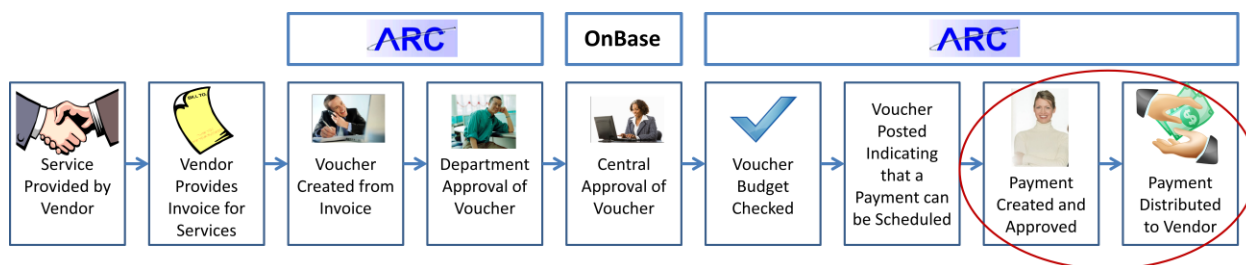
Once a vendor has been created in ARC and the University has transacted with these vendors, invoices are submitted for goods or services provided by the Vendor. From these invoices, or in some cases, Purchase Orders, vouchers can be created in ARC to initiate the payment process. Vouchers can be created manually in ARC or can be processed from interfacing files. The manual process is as follows:



Another way vouchers are generated in ARC is through interface files. Interface files contain voucher data, are loaded into ARC on a regular basis, and are automatically generated into vouchers. One particular type of interface file is one containing P-Card Transactions. P-Card transactions will be loaded into the system via an interface file, reconciled, and generated into vouchers so they can be paid. The general process for Interface Files is as follows:



Once vouchers have been created both manually and through interface files, payments can be scheduled, generated, and distributed to vendors. This completes the Accounts Payable end-to-end process, and is as follows:



JOB AID



Alternatively, payments can be generated for employees/payees needing to be reimbursed for business-related purchases:

